A PHYSIOLOGICALLY-MOTIVATED MODEL OF CREATININE AND FLUID VOLUME DYNAMICS IN ACUTE KIDNEY INJURY

by

Evan D. Richards

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2018

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This thesis was presented

by

Evan D. Richards

It was defended on

March 26, 2018

and approved by

Robert S. Parker, Ph.D., Professor, Department of Chemical and Petroleum Engineering

Gilles Clermont, M.D., Department of Critical Care Medicine

Jason E. Shoemaker, Prodessor, Department of Chemical and Petroleum Engineering

Thesis Advisor: Robert S. Parker, Ph.D., Professor, Department of Chemical and Petroleum Engineering

A PHYSIOLOGICALLY-MOTIVATED MODEL OF CREATININE AND FLUID VOLUME DYNAMICS IN ACUTE KIDNEY INJURY

Evan D. Richards, M.S.

University of Pittsburgh, 2018

Blood serum creatinine concentration (SCr) is used as a surrogate for kidney function. In an intensive care setting, SCr is used to estimate the extent of Acute Kidney Injury (AKI). AKI occurs in a 67% of intensive care unit (ICU) admissions [17], and it can lead to devastating impacts on the body including the development of interstitial and pulmonary edema, toxin accumulation, and excess mortality [8]. Previous research shows the benefits of utilizing an absolute scale for measuring SCr [44] and the necessity to consider the impact of systemic volume changes [32]. The present work develops a biologically-motivated, low-order model of volume and creatinine dynamics that further progresses the understanding of an SCr measurement.

Fluid volume is modeled into three interacting spatial compartments representing blood, interstitial volume, and intracellular volume. The blood compartment is further divided into plasma and liquid contained within the hematocrit (red blood cells). The four compartment creatinine model uses a similar structure to the fluid volume model, but combines the intracellular and interstitial volumes together and re-distributes these volumes between the muscular and non-muscular regions. The rate of creatinine generation is known to decrease during AKI [16, 33, 45]. This understanding motivates the need for a model able to capture the influence of changes in rate of creatinine generation on the creatinine concentrations across the included compartments. A trajectory of creatinine generation rates is included in this work. Simulated studies of dehydration and fluid overload across six days demonstrate the ability of this model to capture kidney function changes in scenarios where absolute SCr

measurement would not recognize AKI as promptly. The calculated differences are shown in hypothetical, clinical scenarios by varying levels of kidney function at differing stages of chronic kidney disease - none, stage 2, stage 3, and stage 4. To examine the applicability of this model for a clinical setting, its performance is tested by studying its aptitude to fit data collected from ten different patients at the University of Pittsburgh Medical Center (UPMC) and portray kidney function.

Patient data fits accompanied by the simulated studies demonstrate the importance of integrating human physiology into a low-order model that considers critical components of volume and creatinine dynamics.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

First and foremost, I want to thank God for giving me this blessing. Through the good times and the bad, you have guided my path, given me peace, and driven my ambitions.

I want to thank my beautiful wife for encouraging me and pushing me to pursue my dreams. You are such a blessing in my life. I couldn't do this without having my best friend by my side. Not only do you support me, but your energy and smile put a smile on my face only you know how to do. Mother and father, you have always taught me to work hard and push forward. Thank you for always being there for me and supporting me through the most trying times of my life. I am so thankful to have parents like you both. My newest mother and father; you both have been there for me since Sarah introduced me to you. I am so blessed! All of my immediate and extended family have supported me through all of this and I can't thank all of you enough. To my friends; thanks for dragging me away from my desk from time to time. The encouragement I receive from all of you is irreplaceable.

Dr. Parker, thank you for opening the doors to a whole world I have never had the opportunity to explore. Your lab and what it does is exactly what I was hoping to learn. I have learned so much. Your knowledge and methodology of advising taught me how interconnected human physiology is with engineering. Thank you for allowing me to take my first steps in coding through your group. Not only did I take my first steps, but I gained a ton of knowledge that I would have never learned on my own. Integrating coding with the medical field was something I always wanted to be a part of. Dr. Clermont, your combined knowledge of medicine, mathematics, and coding allowed me to discover so much. Your time was greatly valued and impacting. The hours you spent teaching me MySQL and how to process data were skills that I hope to use for the rest of my life. To both of you, thank you for your patience at times and your encouragement in many facets of academia. I want to

thank Li Ang Zhang for all of his help in learning python and challenging me to grow. I don't think I would have learned half of what I did if it wasn't for your patience and willingness to teach me. I will never be able to thank you enough for the time you gave me. All of my other lab members: Matt, Michelle, Megan, Flo, Monica, and Peter - You all were there for me at varying times and I want to say thank you so much for your friendship and support.

I would also like to thank UPMC for granting me access to patient data. A great portion of my research would not be possible if it wasn't for the data you provided.

1.0 INTRODUCTION

Serum creatinine (SCr) levels are used to diagnose Acute Kidney Injury (AKI). AKI is defined as an abrupt decline in glomerular filtration rate (GFR) and it causes an upset in both fluid volume and creatinine trafficking. Fluid maintenance is required in a significant number of admissions to the ICU [22]. Since SCr is extracted from a changing fluid volume, it should take into account fluid dynamics . Previous studies have shown a change in creatinine generation rate during AKI [16, 33, 45]. Interpreting SCr measurements should incorporate these two important dynamic factors for drawing a definitive picture of GFR. Integrating these dynamics would also support fluid management during AKI to avoid interstitial (and pulmonary) edema [8] and dehydration.

In 2004, the first consensus definition of AKI was put forward by the "Acute Dialysis Qualitative Initiative" (ADQI) group and the classification was called *RIFLE* (Risk, Injury, Failure, Loss of kidney function, and End-stage kidney disease) [1]. The ADQI group sought to define AKI using the % increase of SCr from standard, baseline levels and the rate of urine output over varying time intervals [1]. Another AKI classification system was established in May of 2007 by the Acute Kidney Injury Network (AKIN) [36]. The intention of AKIN was to categorize AKI into three, distinct stages. Similar to the RIFLE criteria, AKIN used a % increase in SCr over baseline and urine output criteria. AKIN also set SCr thresholds that if surpassed would indicate a degree of AKI [36]. The classifications of RIFLE and AKIN are given in appendix A. In March of 2009, two researchers named S. Waiker and J. Bonventre proposed an alternative definition to AKI using SCr measurements on an absolute scale rather than % change from baseline [44]. Their justification was that % increase in SCr is late to detect AKI for patients pre-diagnosed with Chronic Kidney Disease (CKD). Patients with CKD have an increased baseline level of SCr and the % change in SCr is not

as accurate of a representation of kidney function [44]. A model of volume and SCr response during cardiac arrest was subsequently built [32] and suggests that as body fluid volumes change, so does the concentration of creatinine in the plasma and extravascular space [32].

A novel model embodying the interactions between volume, creatinine generation rate, and SCr trafficking is proposed herein. Using an absolute scale for SCr measurement [44], this model draws upon human physiology by employing kinetic principles to fluid volume trafficking, SCr transport, and creatinine generation rates between anatomical compartments. The volume model seeks to replicate volume regulatory processes at hydration levels extending from dehydration to interstitial edema. Previous studies have found strong correlations between the change in rate of urine output and AKI [23]. These findings solidify the necessity to develop a model capturing volume dynamics in their realistic physiologies. SCr dynamics are volume-dependent and responsive to AKI-induced changes in creatinine clearance.

Simulated studies of dehydration and fluid overload across six days demonstrate the ability of this model to capture kidney function changes in scenarios where absolute SCr measurement and % SCr change would not recognize AKI as promptly. The calculated differences are shown in hypothetical, clinical scenarios by varying levels of kidney function at differing stages of chronic kidney disease - none, stage 2, stage 3, and stage 4. To examine the applicability of this model for a clinical setting, its performance is tested by studying its aptitude to fit data collected from ten different patients at the University of Pittsburgh Medical Center (UPMC) and portray kidney function. Patient data fits accompanied by the simulated studies conclude the importance of integrating human physiology into a trivial, low-order model that considers critical components of volume and creatinine dynamics.

1.1 KIDNEY FUNCTION AND BIOMARKER SELECTION

The kidneys are responsible for urine formation and removal of materials the body does not require. Figure 1.1 gives an overview of kidney function by describing the process of glomerulus filtration, reabsorption of desired materials after filtration, and urinary excretion. The process starts by blood flowing from the afferent arteriole to the glomerular capillaries.

The capillaries filter the blood by removing toxic waste and ions such as potassium and chloride. Targeted solutes and water are then reabsorbed by the peritubular capillaries. The peritubular capillaries also secrete specific materials from the blood via active transport [13]. Tubular secretion is another mechanism of the kidneys and regulates blood volume as well as pressure to maintain and adjust the concentration of solutes in the blood [5, 14]. These processes promote homeostasis throughout the human physiology.

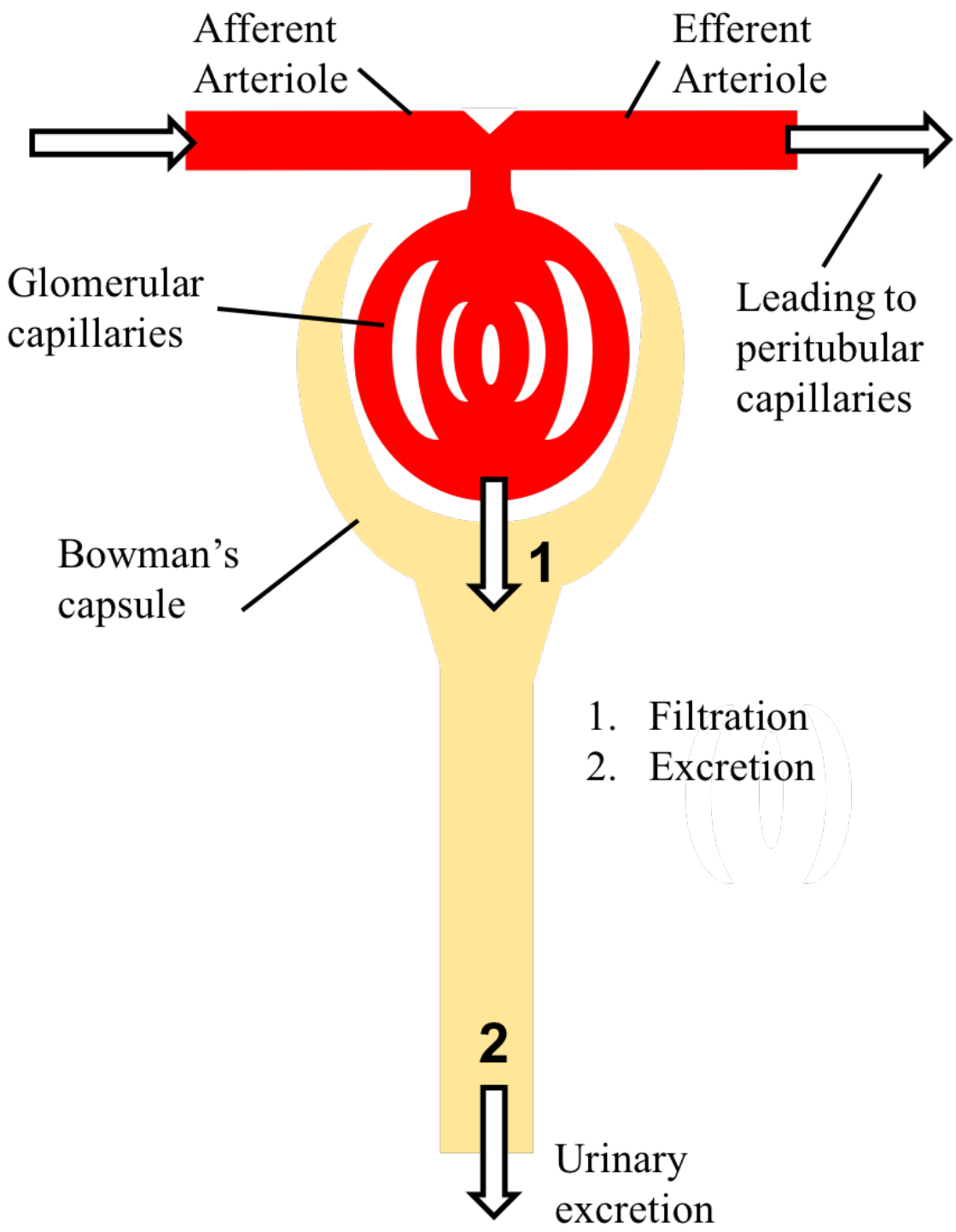

Figure 1.1: General kidney function and description of pathways fluid and molecular trafficking through the kidneys. This figure is an adaptation of the original in Guyton and Hall's Textbook of Medical Physiology [13].

If the substance or liquid is filtered or secreted and not reabsorbed, it becomes a component of the urine composition. Because of these processes, biomarker detection in urine is not ideal for monitoring GFR. Therefore, detection of the biomarkers in the blood serum give a more definitive account of kidney function. The optimal biomarker will be one that is filtered by the glomerulus, not reabsorbed, not secreted, and is not significantly influenced by other organs in the body. Continually, biomarkers are being researched and proposed to advance the diagnosis and treatment of AKI [25, 27, 34]. These biomarkers lack specificity to AKI diagnosis or were found to be only comparative to clinical evaluation or standard laboratory measurements [30]. To this point, the most ideal biomarker has been found to be creatinine measured in the blood serum. This biomarker is currently part of the definition of AKI [1, 36] and the intent of this work is to further improve the usage of SCr and urine output rate as indicators of AKI.

1.2 CURRENT MODELS

Mathematical models are built in an attempt to capture pieces of reality and bring explanation to conjectures in a trivial sense. Previous models have been manufactured to explain AKI and further advance its diagnosis. Sections 1.2.1 and 1.2.2 give descriptions of two current models published and well referenced in this field of study.

1.2.1 Waikar and Bonventre Model

In 2009, Drs. Waikar and Bonventre developed two different models that would aim to describe SCr as a biomarker for predicting AKI. The first was a one compartment model of creatinine and the second was a two compartment model. The two compartment model aimed to show whether multiple compartments would yield significantly different results in SCr measurements. For the one compartment model, the volume was 42 L [44] and constant. So, when there was a change in SCr, this meant creatinine diffused instantaneously throughout the whole body. Figure 1.2 illustrates their first model.

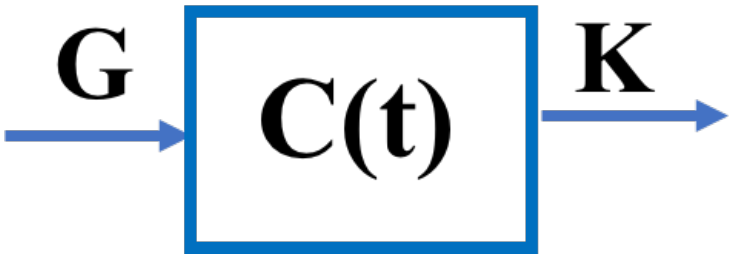

Figure 1.2: Waikar and Bonventre's one compartment model [44].

Equation (1.1) mathematically describes this one compartment space.

$$\frac{dC(t)}{dt} = \frac{(G - KC(t))}{V} \tag{1.1}$$

In the equation above, the system volume (V) is constant at 42 L for a 70 kg patient. Here, creatinine generation (G) and kidney filtration (K) occur in the same compartment [44]. K changes with GFR and G is modified with the degree of CKD. During AKI, G stays constant at its original value. These two constants govern the concentration of creatinine (C).

Figure 1.3 shows their two compartment model. This two compartment structure was built to illustrate the metabolism of creatinine since its generation occurs in the intracellular space (C_i) and diffuses by first-order kinetics into the extracellular regions of the human anatomy (C_e). The volume of the intracellular and extracellular compartments are 28L and 14L, respectively [44].

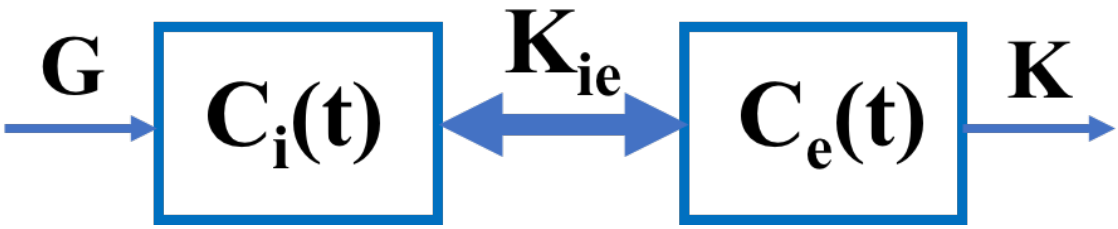

Figure 1.3: Waikar and Bonventre's two compartment model [44].

Equations (1.2) and (1.3) describe Figure 1.3. k_{ie} is the rate of creatinine diffusion between C_i and C_e.

$$\frac{dC_i(t)}{dt} = \frac{G - K_{ie}(C_i - C_e)}{V_i} \tag{1.2}$$

$$\frac{dC_e(t)}{dt} = \frac{K_{ie}(C_i - C_e) - KC_e}{V_e} \tag{1.3}$$

Their analysis concluded a serious flaw in using percentage change in SCr from baseline to diagnose kidney function. They made their point evident by discussing CKD, its influence on baseline SCr, and how any given change in creatinine clearance will yield slower rises in SCr based on the degree of CKD present. This finding drew them to the conclusion that an absolute scale would be more beneficial at diagnosing AKI versus a percentage change. The one compartment model performed similarly to the two compartment model. If the two compartment structure had considered physiological details, increased differences would be seen in their systems.

Waikar and Bonventre made a key finding in arguing the necessity for using an absolute scale. Considering additional items influencing SCr interpretation will also significantly impact the time until diagnosis of AKI.

1.2.2 Pickering Model

In 2013, J.W. Pickering et. al. developed a model that would demonstrate the influence of system volume dynamics on creatinine dynamics [32]. The motivation for their model development was improve prediction of kidney function in patients suffering from cardiac arrest depleted volume levels [32]. Figure 1.4 is a representation of their proposed model.

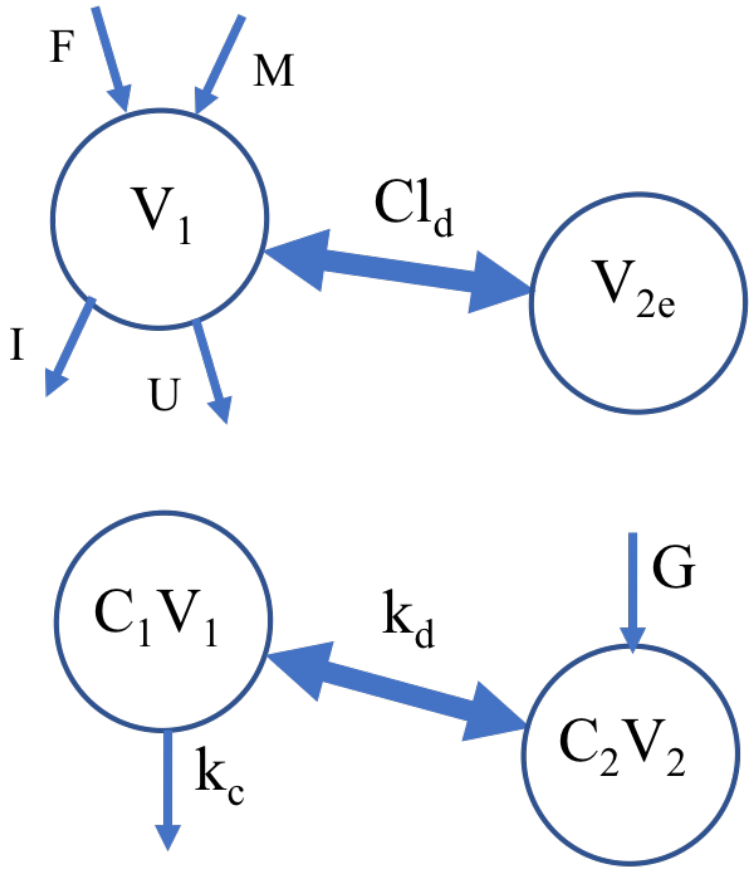

Figure 1.4: A remake of J. W. Pickering's model of volume and creatinine dynamics [32]. The top model is of volume dynamics. The bottom system represents the mass-balance relationship between volume and creatinine.

The top model describes volume dynamics and the other gives the mass-balance relationship between creatinine concentration and volume. The following differential equations were developed to mathematically describe Figure 1.4.

$$\frac{dV_1}{dt} = \frac{dB}{dt} + \frac{dM}{dt} - \frac{dI}{dt} - \frac{dU}{dt} - \frac{dT}{dt} \tag{1.4}$$

$$\frac{dV_{2e}}{dt} = \frac{dT}{dt} \tag{1.5}$$

$$\frac{dC_1}{dt} = \frac{k_d(C_2V_2 - C_1V_1) - k_rC_1V_1 - C_1\frac{dV_1}{dt}}{V_1} \tag{1.6}$$

$$\frac{dC_2}{dt} = \frac{\frac{dG}{dt} + k_d(C_1V_1 - C_2V_2) - C_2\frac{dV_2}{dt}}{V_2} \tag{1.7}$$

V_1 and C_1 are the volume and creatinine concentration in the blood plasma compartment, respectively. Likewise, V_{2e} and C_2 are the volume and creatinine concentration in the extravascular compartment. The initial volumes used by Pickering et. al. for compartments V_1 and V_{2e} at t = 0 were 2.8L and 8.4L, respectively for a 70 kg hypothetical patient [32]. In Equation (1.4), $\frac{dB}{dt}$ is the liquid infusion rate, $\frac{dM}{dt}$ is the metabolic fluid production, $\frac{dI}{dt}$ is insensible fluid loss, and $\frac{dU}{dt}$ is the urine production rate. The net rate of fluid exchange between compartments V_1 and V_{2e} is governed by Equation (1.8).

$$\frac{dT}{dt} = Cl_d\left(\frac{(v_1 - V_{1,0})}{V_{1,0}} - \frac{(v_{2e} - V_{2e,0})}{V_{2e,0}}\right) \tag{1.8}$$

Here, v_1 and v_{2e} are measurements of the expanded compartment volumes and $V_{1,0}$ and $V_{2e,0}$ are the initial volumes of those compartments [32]. Cl_d represents the rate of distribution clearance between the two compartments and was labeled in Figure 1.4 by Pickering et. al. [32]. This equation demonstrates that as either volume expands or contracts, $\frac{dT}{dt}$ will adjust to proportionally distribute fluid volume between the two compartments. This equation grants boundless increase and decrease in volume within both the vascular and extravascular compartmental spaces. Equation (1.9) expresses the rate of creatinine distribution (k_d) between the two creatinine compartments.

$$k_d = \frac{G_0}{C_0(V_{2,0} - V_{1,0})} \tag{1.9}$$

G_0 is the initial rate of creatinine generation and $V_{2,0}$ and $V_{1,0}$ are the initial volumes of the vascular and extravascular compartments [32]. Equations (1.10) through (1.14) govern the rates of urine production and creatinine filtration. Equation (1.11) expresses the reabsorption rate of fluid after kidney filtration. The reabsorption rate is influenced by the level of

hydration of the patient and is given as a piecewise function. The rate of creatinine filtration is given by Equation (1.12) and is a product of the initial rate of creatinine generation as well as the rate of glomerular filtration (GFR) as it is subject to change over time as indicated by Equation 1.14. Their model depicts how GFR is influenced by kidney function [32]. If the kidneys lose 50% of their functionality, GFR decreases by 50%.

$$\frac{dU}{dt} = GFR(t)\frac{(100 - Reabs\_rate)}{100} \tag{1.10}$$

$$Reabs\_rate = \begin{cases} 99.7e^{-0.0000109(Hydration-90)^{2.81}} & < 100\% Hydration \\ 1.30x10^{-9}(120 - Hydration)^{7.71} + 85 & \geq 100\% Hydration \end{cases} \tag{1.11}$$

$$k_r(t) = k_{r0}(1 - \Delta g(GFR(t))) \tag{1.12}$$

$$k_{r0} = \frac{g_0}{C_0 V_{1,0}} \tag{1.13}$$

$$\Delta g(GFR(t)) = \frac{GFR(0) - GFR(t)}{GFR(0)} \tag{1.14}$$

The model designed by Pickering et. al. accounts for more dynamics then the model proposed by Waikar and Bonventre. Both models have enhanced the interpretation of SCr. The model proposed within this body of work aims to build upon the understanding of SCr by including more physiological details of volume and creatinine dynamics. This model will concurrently include biological equations to fluently describe biological phenomena.

1.3 MODELING OF VOLUME AND CREATININE DYNAMICS

Section 1.2 identified two prominent models used to improve the diagnosis of AKI. More foundational detail is given here that influenced the design of the model in this work.

1.3.1 Creatinine Characteristics

Creatinine is a cyclic molecule yielded by both creatine and phosphocreatine in the human anatomy. It has both a positive and negative charge making it a zwitterion. Zwitterions won't interact with the surface of non aqueous proteins or antibodies in the blood. Due to its low interaction with other materials, SCr measurements will effectively represent the

total amount of creatinine within the blood. Zwitterions are water soluble and will diffuse throughout all major volume compartments [6, 39].

1.3.2 Volume Dynamics

Volume dynamics throughout multiple compartments of the human body should be accounted for. AKI and total fluid volume directly impact the urinary production rate. If AKI is not detected, fluid maintenance will continue under the assumption of nominal kidney function. If the rate of fluid input is greater than fluid output, fluid volume will begin to accumulate. Vascular compliance has a maximum threshold at 7L. Accumulated fluid will be pushed to the extravascular compartments as this limit is approached. Fluid accumulation in the interstitial compartment may yield pulmonary edema if the rate of fluid input is not decreased [13]. Dehydration will also influence SCr measurements due to a change in total fluid volume.

A distinction is made between the blood plasma compartment and red blood cells. SCr measurements come from only the blood plasma. The concentration of creatinine in red blood cells is not consistently the same as blood plasma due to the slow rate of volume and creatinine diffusion between the two compartments [13, 39]. Generated creatinine will enter the vascular compartment. Assuming the generated creatinine enters a 5 L whole blood volume, the rate of SCr increase during AKI will be delayed by 40% compared to a 3 L blood plasma compartment. Because of these conditions, a distinction should be made between the volume of blood serum and red blood cells.

The general understandings listed here drive the development of a four compartment volume model. Accounting for volume distribution would give more reliable diagnosis from a SCr measurement and alert the potential for interstitial edema. Human physiology will further direct the dynamic relationships between these four compartments in chapter 2.

1.3.3 Creatinine Concentration Dynamics

Creatinine will diffuse throughout the human anatomy, but the rate is not instantaneous [6, 39]. Due to the slower rate of diffusion between subsections and in an effort to effectively

capture creatinine dynamics, there are three major compartments considered for designing the creatinine model. The compartments are plasma, red blood cells, and the extravascular region.

Creatinine is generated in muscle located in the extravascular compartment. Muscle does not comprise the entire space so creatinine should not diffuse directly into the entire extravascular region. This subdivides the extravascular compartment into two, distinct compartments – muscular and non-muscular. Previous studies have given reason to assume creatinine will diffuse from the muscular space to the vascular compartment proceeding generation. The interaction of the vascular and muscular compartments from a physiological standpoint would further conclude this assumption. Creatinine generation is not constant during AKI [16, 45] and will significantly influence creatinine dynamics as it changes. A SCr measurement should account for the change in creatinine generation.

Chapter 2 will elaborate upon these details at greater length.

1.4 THESIS OVERVIEW

The structure of the remainder of this work will be presented in the following order. Chapter 2 gives a thorough understanding of model development. Since creatinine dynamics are highly dependent upon volume dynamics, the first section builds the volume model. The volume model is not static and the dynamics need to be grounded physiologically. Therefore, mathematical equations are given to key rate constants. The next section of chapter 2 captures the creatinine dynamics and is presented with similar structure to the volume section. The final section of this chapter compares the developed model to a static system in hypothetical, clinical scenarios. Chapter 3 focuses on the ability of the model to fit data given by the University of Pittsburgh Medical Center (UPMC). The chapter will first discuss the motivation for fitting the model to real data. The following section will elaborate upon the methodology followed for selecting the patients. Proceeding patient selection, the data is then processed for fitting. Patient data fits are then given and discussed. Chapter 4 concludes the thesis with a summary and recommendations for future work.

2.0 MODEL DEVELOPMENT AND ANALYSIS

A compartmental approach was used to develop a mass balance model of creatinine and volume dynamics. The models assume the patient to be 70 kg. Kinetic rates link compartments and are grounded in physiological ranges when possible. The intent is to mimic these physiological responses of fluid volume and creatinine concentrations to replicate actual effects encountered in the ICU due to changes in kidney function. Section 2.1 discusses the synthesis of the model for volume dynamics, the associated dynamic parameters, and how the model structure and equations were selected. Section 2.2 does the same for creatinine dynamics. Section 2.3 applies this model to a set of theoretical, clinical scenarios. The purpose of these scenarios is to illustrate the ability of this model to predict AKI when absolute SCr measurements are incapable of doing so.

2.1 VOLUME DYNAMICS

Previously, one and two compartmental models have been used to demonstrate volume and/or creatinine dynamics between the vascular and extravascular spaces [32, 44]. However, there are volume dynamics in the extravascular space that will influence the volume dynamics of the vascular region. In order to effectively capture interstitial edema, the extravascular compartment ($V_x(t)$) is divided between intracellular ($V_i(t)$) and interstitial ($V_e(t)$) spaces. $V_i(t)$ will account for fluid volume in all cells except red blood cells ($V_{rbc}(t)$). Blood plasma/serum ($V_s(t)$) and $V_{rbc}(t)$ comprise the vascular compartment ($V_b(t)$). A schematic of this model is shown in Figure 2.1. Fluid can exit the model three different ways. Interstitial compartment fluid loss (E) can occur, however, it is not incorporated into the theoretical

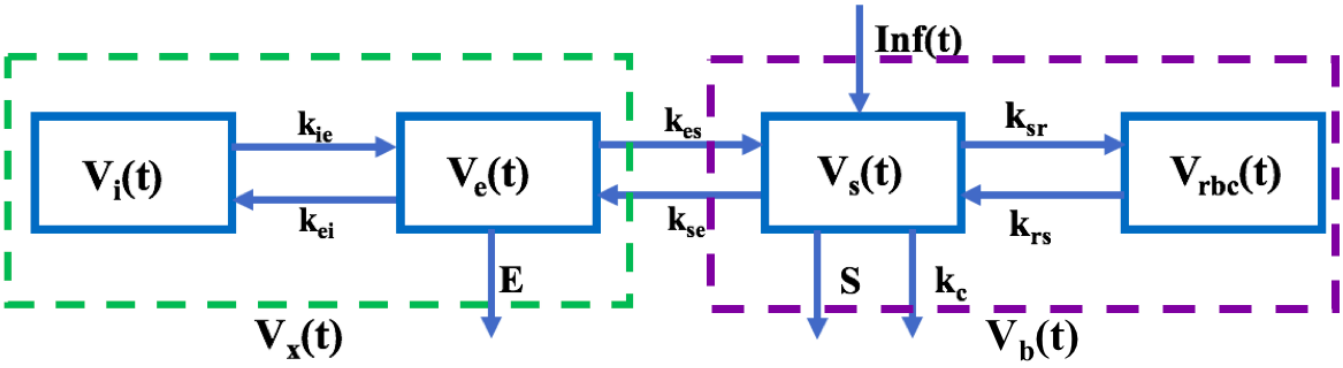

Figure 2.1: A four-compartment model illustrating the distribution of volume in a simulated patient.

model. Its loss will occur as a square wave in chapter 3. Fluid output via urine production rate (k_c) and unexplained, additional losses (S) occur by $V_s(t)$. The four-compartment structure is characterized by the following differential equations:

$$\begin{aligned}
\frac{dV_s(t)}{dt} &= k_{rs}V_{rbc}(t) - k_{sr}V_s(t) + k_{es}(V_e(t), V_b(t))V_e(t) - k_{se}(V_e(t), V_b(t))V_s(t) \\
&\quad -k_c(V_{tot}, GFR_s(t)) - S(V_{tot}(t)) + Inf(t) \qquad (2.1) \\
\frac{dV_{rbc}(t)}{dt} &= k_{sr}V_s(t) - k_{rs}V_{rbc}(t) \qquad (2.2) \\
\frac{dV_e(t)}{dt} &= k_{ie}(V_i(t))V_i(t) - k_{ei}(V_i(t))V_e(t) + k_{es}(V_e(t), V_b(t))V_e(t) \\
&\quad +k_{se}(V_e(t), V_b(t))V_s(t) \qquad (2.3) \\
\frac{dV_i(t)}{dt} &= k_{ei}(V_i(t))V_e(t) - k_{ie}(V_i(t))V_i(t) \qquad (2.4)
\end{aligned}$$

Compartments $V_s(t)$ and $V_{rbc}(t)$ were used in place of $V_b(t)$ due to the diffusion of fluid volume and creatinine across the erythrocyte membrane not being instantaneous [39]. Extravascular compartments are dynamic volumes that provide a source (in dehydration) or sink (in fluid overload) for fluid *in vivo* entering the vascular compartment.

2.1.1 Dynamic Volume Parameter Relationships

Here, we discuss the parameters used to describe dynamic relationships between the volume compartments. To obey physiological constraints under dynamic conditions, some parameters were defined using mathematical expressions.

Urine production rate was determined based on the daily rate of fluid intake and excretion required to achieve volume equilibrium. This model considers the rate of fluid intake to be 2.3 L day^{-1}. Fluid is delivered to the model by a bolus and/or continual infusion depending on the condition of the patient. Parameter values not extracted from literature were set to make the model match expected physiological states at equilibrium. At 100% hydration, fluid is driven between compartments by osmotic forces. However, as volume levels shift toward dehydration or edema, additional forces come in to play. With interstitial edema, there is an increase in capillary hydrostatic pressure resulting from a loss of blood vessel compliance as the vessels reach their maximum expansion [13]. Increased fluid flow to $V_e(t)$ is also due to an increase in the permeability of blood capillaries [13]. As fluid levels move toward the state of dehydration, hormonal factors are activated in the renin-angiotensin-aldosterone system (RAAS). RAAS regulates blood pressure within the body by releasing hormones such as renin and angiotensin I. Angiotensin I is converted to Angiotensin II and causes vasoconstriction to occur in order to maintain systemic blood pressure homeostasis. Renin maintains increased salt concentrations within the vascular compartment to keep essential fluid volume for blood flow [13]. As dehydration progresses, the renal system begins to reabsorb fluid and ions after kidney filtration which causes k_c and S to decrease. Maximum and minimum rates of urine production were found in literature for a theoretical 70 kg patient [13]. The resultant expression is given by Equation (2.5) and fit to the digitized scatter plot in Figure 2.2. There are a total of three main contributors to the rate of urine production - tubular reabsorption, glomerular filtration rate, and oliguria [37]. This model does not account for oliguria. Though oliguria is considered a strong indicator of AKI and predictor of mortality [23], its rate of occurrence in the ICU is not as common as the others [37]. The equation for urine output rate is given, below.

$$k_c(GFR_s(t), V_{tot}(t)) = 0.009 + GFR_s(t)\frac{0.833 - 0.009}{1 + e^{-k_{c,slope}(V_{tot}(t) - k_{c,vol})}} \quad (2.5)$$

A study was conducted on humans in which the rate of urine production was measured based on the level of hydration observed [19]. The given graph was digitized and the resulting scatter plot was used to partly yield the equation determined for k_c. $GFR_s(t)$ is a representation of GFR performance on a scale from zero (no glomerular filtration) to one (nominal kidney filtration). As shown in Equation (2.5), the rate of urine production is regulated by $GFR_s(t)$ and the total volume of fluid ($V_{tot}(t)$). Figure 2.2 shows this equation and how it captures the change in urine production rate as the level of hydration in a patient changes. Parameters $k_{c,vol}$ and $k_{c,slope}$ were fit by minimizing the sum of squared error of the data points shown Figure 2.2. The goodness of the fit was analyzed by calculating the sum of squared error (SSE, eq. 2.7), the mean squared error (MSE, eq. 2.7), and the root mean squared error (RMSE, eq. 2.8).

$$SSE = \sum_{n=0}^{n,max-1} (y_n - f(x_n))^2 \quad (2.6)$$

$$MSE = \frac{1}{N} \sum_{n=0}^{n,max-1} (y_n - f(x_n))^2 \quad (2.7)$$

$$RMSE = \sqrt{MSE} \quad (2.8)$$

The data point (n) is represented at y_n, $f(x_n)$ is the predicted value for each data point, and N is the total number of data points. Here, SSE = 0.0096 L^2 h^{-2}, MSE = 0.0002 L^2 h^{-2}, and RMSE = 0.015 L h^{-1}. This is a very low value for RMSE and allows for minimal error when projecting urine output rate using Equation (2.5).

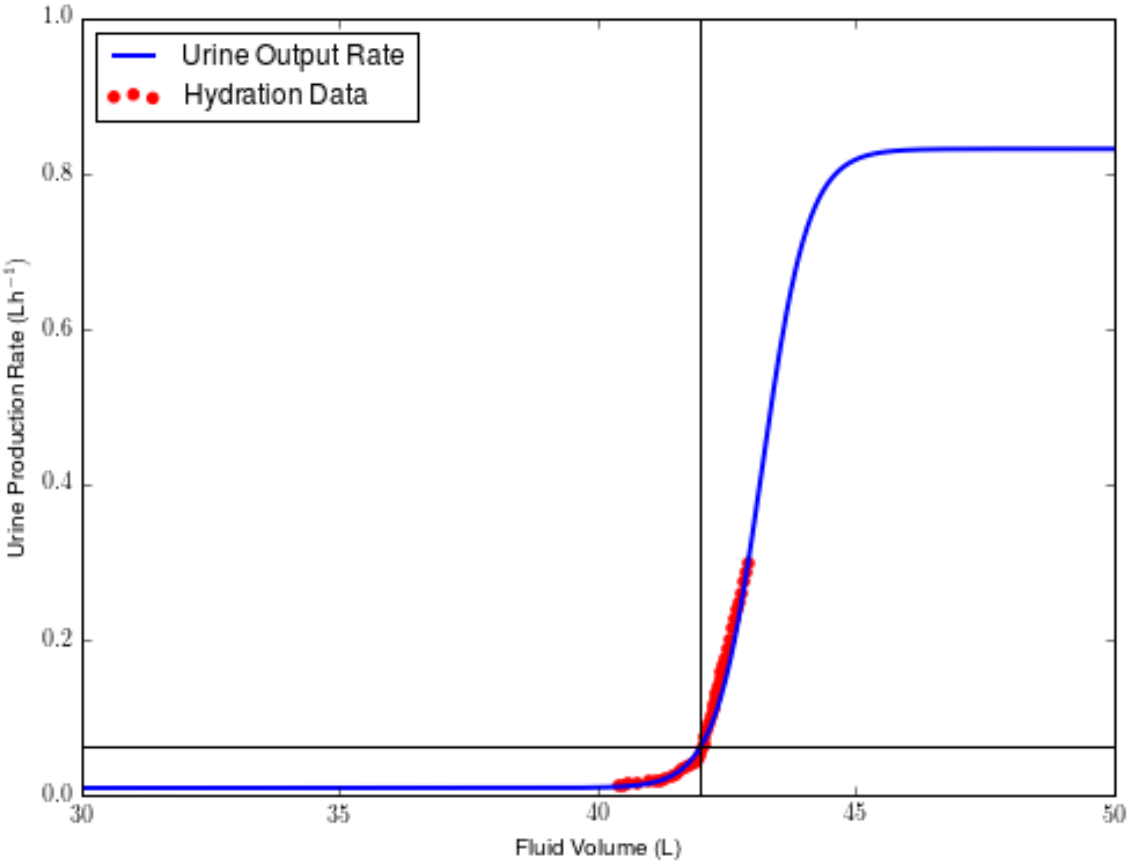

Figure 2.2: Digitized plot of urine production rate (red) and the graphed expression for k_c. The rate of urine production at 100% fluid hydration is marked by two black, solid lines. The proposed equation is graphed by a solid blue line.

The final form of the equation for k_c is given by Equation (2.9).

$$k_c(GFR_s(t), V_{tot}(t)) = 0.009 + GFR_s(t)\frac{0.833 - 0.009}{1 + e^{-2.25(V_{tot}(t) - 43.189)}} \tag{2.9}$$

The maximum rate of urine output is governed by the maximum rate of blood flow through the kidneys. Previous studies have determined an approximate maximum rate of urine production for a 70 kg individual [13]. The minimum rate of urine production is never zero because urine is formed by two pathways - tubular secretion and kidney filtration. If

kidney function is at 0%, tubular secretion will continue to produce urine [13]. The states of dehydration have been defined [40] and Table 2.1 provides the type of dehydration associated with the % total hydration and total volume associated with a 70 kg individual.

Table 2.1: States of dehydration for a 70 kg adult [40].

State of Dehydration	Percentage from 42 L (%)	Total Fluid Volume (L)
Homeostasis	$\leq$3	$V_{tot}(t) \geq 40.7$
Mild to Moderate	3 - 9	$38.2 \leq V_{tot}(t) \leq 40.7$
Severe	$\geq$9	$V_{tot}(t) \leq 38.2$

This definition will be assumed throughout the rest of the work. As the body approaches this state of dehydration, its rate of perfusion will decrease [13] but only to an approximate, standard minimum.

Additional body fluid loss, such as sweat, is accounted for in Equation (2.10). Unlike k_c, S does not consider kidney function because the rate of perfusion is not influenced by it. The equation for S is built to maintain constant fluid loss except during severe dehydration. Assuming the body aims to maintain fluids during severe dehydration, S shuts off at $V_{tot}(t) \leq 38.2$ L. Equation (2.10) gives the equation for S.

$$S(V_{tot}(t)) = \frac{0.1/3}{1 + e^{-6(V_{tot}(t)-38.2)}} \tag{2.10}$$

Both k_c and S have built-in biological switches to account for total body fluid levels as they approach maximum and minimum thresholds of fluid excretion. The biological switch is a logistic equation. The logistic equation postulates that relative rates of growth will decrease as they approach a limiting factor. The limiting factor for k_c is the maximum rate of blood flow through the kidneys [13]. The logistic equation is commonly used to explain biological rates [2, 12] and is well-suited for this model.

Blood volume ($V_b(t) = V_s(t) + V_{rbc}(t)$) has a maximum physiological value at the capacitance limit of the blood vessels. For a 70 kg individual the volume limit is 7 L [13]. The model was able to achieve this by incorporating a biological switch into Equation (2.11) that allows linear volume increases proportional to the increase of $V_e(t)$ up to 7 L, after which additional volume is forced into the extravascular space. The physiological constraints governing volume dynamics between $V_s(t)$ and $V_e(t)$ are captured by rates k_{se} and k_{es}. Equation (2.11) gives the equation for k_{se}. k_{es} is a constant calculated to give system homeostasis at 100% hydration.

$$k_{se}(V_e(t), V_b(t)) = \frac{44}{3} + \frac{2.14V_e(t)}{1 + e^{-5(V_b(t)-6.8)}} \tag{2.11}$$

The fraction $\frac{44}{3}$ in Equation (2.11) is a product of $V_e(t)$, $V_s(t)$, and k_{es}. Their respective values are given in Table 2.2 and will be discussed later in greater detail.

The maximum increase in volume for a cell is approximately 50% beyond its size at equilibrium [24, 29]. $V_i(t)$ accounts for all cells outside the vascular compartment. Due to osmotic effects, the observed volume increase in $V_i(t)$ is further limited to 10% its value at equilibrium. Equations for k_{ie} and k_{ei} translate the physiological constraints of $V_i(t)$ into functional form with a biological switch. If $V_i(t)$ and $V_b(t)$ are at maximum volume capacity, fluid will only accumulate in the interstitial compartment.

$$k_{ie}(V_i(t)) = \frac{5}{1 + e^{3(V_i(t)-26)}} \tag{2.12}$$

$$k_{ei}(V_i(t)) = \frac{26}{11} k_{ie} \tag{2.13}$$

The volume dynamics driving the diffusion of fluid between $V_{rbc}(t)$ and $V_s(t)$ were determined in previous studies [39]. Equation (2.14) shows this parameter and k_{sr} was determined to be $0.006 h^{-1}$.

$$k_{rs} = \frac{3}{2} k_{sr} \tag{2.14}$$

Figure 2.3 demonstrates the model's ability to capture system homeostasis during nominal kidney function and standard fluid intake. This figure shows a 70 kg individual consuming 2.3 L of fluid per day. The individual is awake 16 hours per day. While awake, the individual will consume 0.2875 L once every 2 hours. The fluid is received by the model via bolus addition to $V_s(t)$.

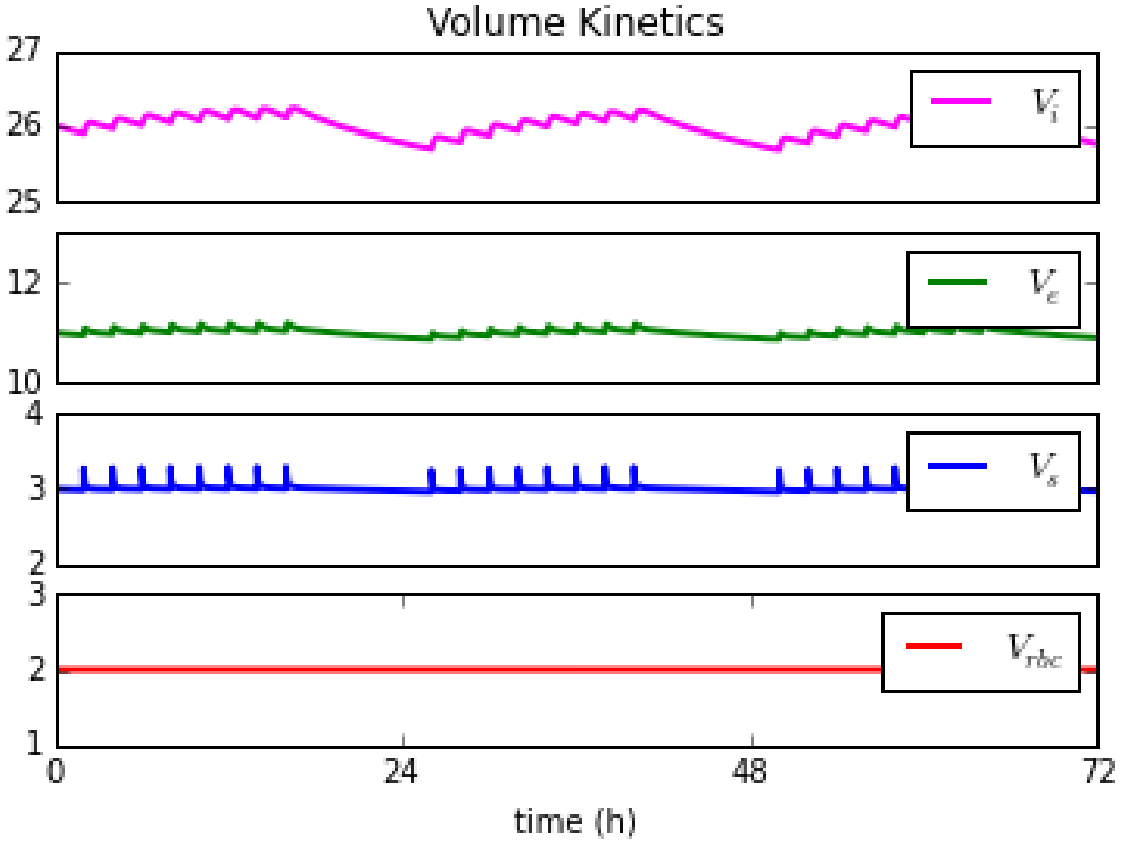

Figure 2.3: Fluid distribution to each of the four volume compartments during standard fluid intake, 100% hydration, and nominal kidney function. Fluid is given to V_s by bolus.

Fluid can also be given to the model via continuous infusion of fluid as would be done in a hospital setting. During nominal kidney function and 100% hydration, the volumes will become static. This static condition is important for demonstration of the model's ability to achieve equilibrium in all compartments. The values of the parameters during equilibrium are shown in Table 2.2. Some of these values were given by literature while others were calculated.

Table 2.2: Rate and volume parameters for the 70 kg simulated patient at model equilibrium.

Parameter	Value	Units	Source
k_{sr}	0.006	h^{-1}	[39]
k_{rs}	0.009	h^{-1}	Calculated
k_{se}	14.67	h^{-1}	Calculated
k_{es}	4.00	h^{-1}	[13]
k_{ei}	11.82	h^{-1}	[13]
k_{ie}	5.00	h^{-1}	Calculated
Initial Condition	Value	Units	Source
V_s	3	L	[13]
V_{rbc}	2	L	[13]
V_e	11	L	[13]
V_i	26	L	[13]

The volumes given in the table are approximate values determined by previous studies [13].

2.2 CREATININE CONCENTRATION DYNAMICS

The creatinine kinetics model is coupled to the volume kinetics model. Because creatinine is a zwitterion it absorbs into extravascular fluid [39]. Its rate of diffusion between the main regions of the body is less than true ions such as urea [39]. Understanding these rates will give a clearer interpretation of SCr. The creatinine concentration model has four compartments: plasma ($C_s(t)$), red blood cell ($C_{rbc}(t)$), muscular ($C_{xm}(t)$) and the

remaining extravascular space ($C_{xe}(t)$). Studies have shown that creatinine does not bind to proteins [7, 39], and therefore all creatinine is processed by the kidneys. The nonvascular compartments embody both $V_i(t)$ and $V_e(t)$ compartments as shown in Figure 2.1. The distribution of fluid between volume compartments $C_{xm}(t)$ and $C_{xe}(t)$ was determined by previous work stating the body is approximately 40% muscle in a healthy individual [21]. Creatinine levels in red blood cells are not quantified by SCr measurements, however, these must be accounted for within the model as creatinine does distribute into red blood cells at a given rate [39]. Therefore, creatinine concentrations are modeled via dynamic equilibration across the erythrocyte membrane. Figure 2.4 shows the compartmental structure of the creatinine concentration model. Volumes from section 2.1 associated with each compartment are included in the figure.

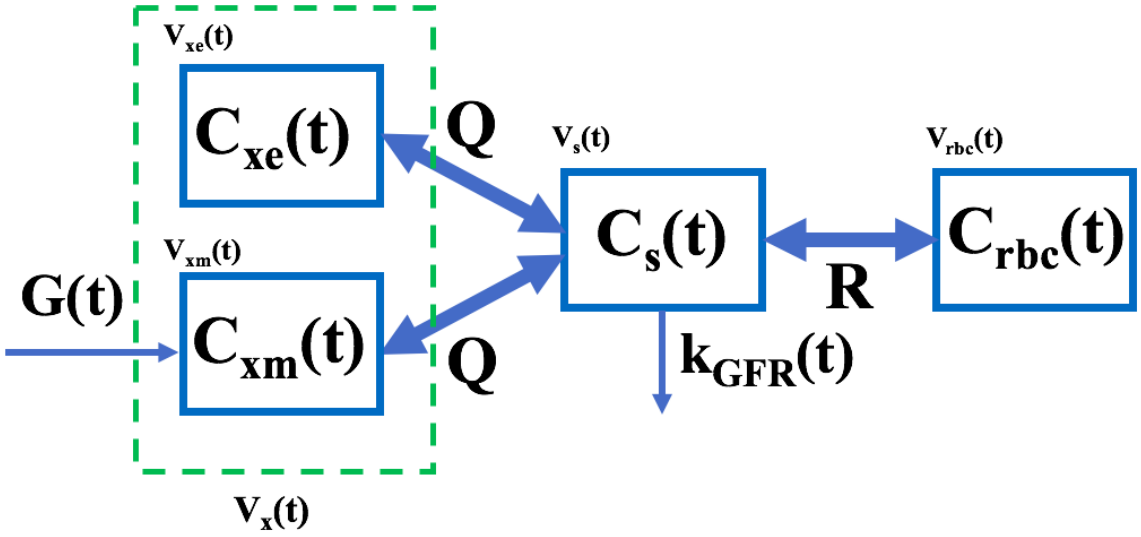

Figure 2.4: A four-compartment model describing creatinine distribution in a simulated patient. The associated volumes are given adjacently to the respective creatinine compartment.

Creatinine generation ($G(GFR_s(t))$ is a product of muscle metabolism [38, 41]. Therefore, $G(GFR_s(t)$ occurs in $C_{xm}(t)$ and not to the whole, extravascular space.

Creatinine diffuses from $C_{xm}(t)$ to $C_s(t)$ since the flow of blood through the muscular compartment acts as a sink for creatinine that accumulates in $C_{xm}(t)$. Previous models

show the diffusion of creatinine throughout the whole extravascular region as soon as it is generated [32]. Other research has simultaneously monitored creatinine interstitial and serum creatinine concentrations [6]. Their works shows $C_{xe}(t)$ trails $C_s(t)$ when creatinine levels are not held at equilibrium [6]. For this reason, Creatinine diffuses from $C_{xm}(t)$ to $C_s(t)$ and then to compartments $C_{xe}(t)$ and $C_{rbc}(t)$.

Mathematically, the creatinine concentration model can be represented as follows:

$$V_{xm}(t) = 0.4(V_i(t) + V_e(t)) \tag{2.15}$$

$$V_{xe}(t) = 0.6(V_i(t) + V_e(t)) \tag{2.16}$$

$$\frac{dC_{xe}(t)}{dt} = \frac{Q(C_s(t) - C_{xe}(t))}{V_{xe}(t)} - \frac{C_{xe}(t)}{V_{xe}(t)}\frac{dV_{xe}(t)}{dt} \tag{2.17}$$

$$\frac{dC_{xm}(t)}{dt} = \frac{G(GFR_s(t))}{V_{xm}(t)} + \frac{Q(C_s(t) - C_{xm}(t))}{V_{xm}(t)} - \frac{C_{xm}(t)}{V_{xm}(t)}\frac{dV_{xm}(t)}{dt} \tag{2.18}$$

$$\begin{aligned}\frac{dC_s(t)}{dt} = {} & \frac{Q(C_{xm}(t) + C_{xe}(t) - 2C_s(t))}{V_s(t)} + \frac{R(V_{rbc}(t), V_s(t))(C_{rbc}(t) - C_s(t))}{V_s(t)} \\ & -k_{gfr}(GFR_s(t))C_s(t) - \frac{C_s(t)}{V_s(t)}\frac{dV_s(t)}{dt}\end{aligned} \tag{2.19}$$

$$\frac{dC_{rbc}(t)}{dt} = \frac{R(V_{rbc}(t), V_s(t))(C_s(t) - C_{rbc}(t))}{V_{rbc}(t)} \tag{2.20}$$

From the volume model, extravascular volumes are given by $V_{xm}(t)$ and $V_{xe}(t)$. k_{gfr} represents glomerular filtration rate. At nominal kidney function, the rate of creatinine generation is equivalent to the rate at which creatinine is cleared. Rate constants Q and R capture the gradient-based transport of creatinine between the plasma and extravascular, or plasma and red blood cell, compartments, respectively. At equilibrium with nominal kidney function, the creatinine concentrations in $C_s(t)$, $C_{rbc}(t)$, and $C_{xe}(t)$ are equal. $C_{xm}(t)$ is greater in creatinine concentration since creatinine generation occurs in this compartment. If the rate constant Q was to significantly increase, $C_{xm}(t)$ would approach the concentrations in the other 3 compartments. The rates of creatinine distribution for $C_{xm}(t)$ and $C_{xe}(t)$ are identical as the rate of transport of creatinine between the vascular and extravascular compartments should be relatively equivalent [6]. Table 2.3 lists the parameter values at fluid volume and creatinine concentration homeostasis.

2.2.1 Dynamic Creatinine Parameter Relationships

Under dynamic conditions, parameters from Table 2.3 will have their own mathematical relationships. Equation (2.21) was incorporated from previous research [39]. Simple diffusion governs the equilibrium of creatinine between compartments $C_{rbc}(t)$ and $C_s(t)$. The functionality for R was determined experimentally to be dependent on relative compartment volumes, water fractions, hematocrit, and overall blood volume [39].

$$R(V_{rbc}(t), V_s(t)) = A\frac{f_{ew}V_{rbc}(t)}{(\frac{f_{ew}}{f_{pw}})\frac{V_{rbc}(t)}{V_s(t)-V_{rbc}(t)}+1} \tag{2.21}$$

Schneditz et. al. derived Equation (2.21) within their work [39]. They built a two compartment model that would represent the diffusion of creatinine between erythrocytes and blood plasma. In vitro, they recorded the movement of creatinine from red blood cells to blood plasma and from blood plasma to red blood cells. The water fractions in their paper for plasma (f_{pw}) and erythrocytes (f_{ew}) are 0.93 and 0.7, respectively. The rate constant A was experimentally determined to be 3.12 h^{-1} +/- 0.78 h^{-1}. Schneditz et. al. concluded the rate of diffusion between the compartments would be dependent upon compartment volumes and rate constants experimentally determined. Their work also showed that creatinine diffused at a slower rate than other water soluble compounds such as urea. The final form of Equation (2.21) is given in Equation (2.22).

$$R(V_{rbc}(t), V_s(t)) = 3.12\frac{0.7V_{rbc}(t)}{(\frac{0.7}{0.93})\frac{V_{rbc}(t)}{V_s(t)-V_{rbc}(t)}+1} \tag{2.22}$$

The rate of glomerular filtration ($k_{gfr}(GFR_s(t))$) is derived from the generation rate of creatinine (44 mg h^{-1}), $V_s(t)$, and $C_s(t)$ at system homeostasis and $GFR_s(t) = 1$. The initial $V_s(t)$ and $C_s(t)$ values are given in Tables 2.2 and 2.3.

$$k_{gfr}(GFR_s(t)) = \frac{44}{V_{s0}C_{s0}}GFR_s(t) \tag{2.23}$$

Notice $k_{gfr}(GFR_s(t))$ is only dependent upon changes in kidney function. This is due to the kidney's ability to autoregulate itself over a large range of volumes. Therefore, the rate of glomreular filtration is constant. There are extreme conditions of blood pressure and blood volume that could impact GFR. But, those conditions are outliers and beyond the scope of this work.

Multiple studies have documented the significance that creatinine generation rate plays in diagnosing AKI [16, 31, 32, 33, 45]. Previous studies have shown a decrease in $G(GFR_s(t))$ ranging from 25 to 50% [3, 4, 31, 45]. However, the current definition of AKI does not consider this phenomenon and current models do not include its impact as the glomerular filtration rate varies with a patient in the ICU [33]. Dr's Waikar and Bonventre gave rates of creatinine generation for normal 100% kidney function and stages 2, 3, and 4 of CKD [44]. Here, we use those reference points to build an expression for varying $G(GFR_s(t))$ with changes in kidney function. The values given are $G(GFR_s(t)) = (34.63, 38.00, 41.67, 44.00)$ corresponding to scalar GFR measurements of $GFR_s = (0.17, 0.33, 0.67, 1.00)$, respectively. Equation (2.24) is the expression to be optimized. The minimum rate of $G(GFR_s(t))$ is 30 mg h^{-1} and allows for a 32% decrease in $G(GFR_s(t))$ from 44 mg h^{-1} at $GFR_s(t) = 1$.

$$G(GFR_s(t)) \quad = \quad 30.00 + G_{max}\frac{GFR_s(t)}{(K + GFR_s(t))} \tag{2.24}$$

The parameters to be fit in Equation (2.24) are G_{max} and K. G_{max} is the total change in $G(GFR_s(t))$ from $GFR_s(t) = 0$ to $GFR_s(t) = \infty$. The constant K is the value of $GFR_s(t)$ marking the halfway increase increase from $G(GFR_s(t)) = 30$ mg h^{-1} to $G(GFR_s(t)) = 30$ mg h^{-1} + G_{max}. Fitting the parameters in Equation (2.24) by minimizing SSE yields Equation (2.25).

$$G(GFR_s(t)) \quad = \quad 30.00 + 22.96\frac{GFR_s(t)}{(0.6937 + GFR_s(t))} \tag{2.25}$$

The goodness of the fit was calculated and the values are SSE = 0.030 mg^2 h^{-2}, MSE = 0.008 mg^2 h^{-2}, and RMSE = 0.089 mg h^{-1}. The RMSE indicates the equation has minimal deviation from the provided data points. Equation (2.25) is illustrated in Figure 2.5 and overlays the data points used for its optimization.

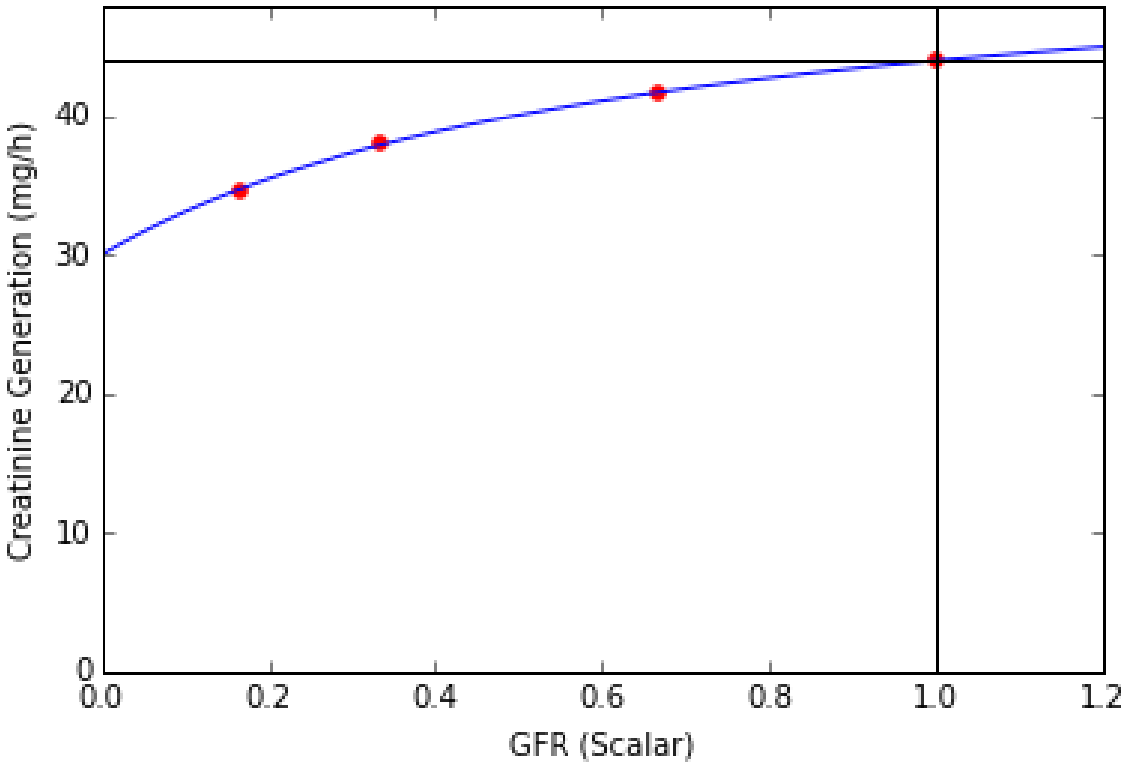

Figure 2.5: Equation for $G(GFR_s(t))$ with data points used for its fitting. The horizontal and vertical black lines identify creatinine generation rate at $GFR_s = 1$.

A modified form of a first order Hill equation was used to build $G(GFR_s(t))$. The Hill equation is continually used to express the rates of biological processes that saturate at both high and low values [9, 35]. This model explains that human physiology will generate creatinine even when $GFR_s(t)=0$. Therefore, a constant is given before the Hill equation. At nominal kidney function, $G(GFR_s(t))$ will produce 44 mg h^{-1} as shown in previous work [44].

The rate of diffusion between the vascular and extravascular spaces (Q) was determined by considering the Starling hypothesis. The hypothesis describes the rate of fluid movement across capillary membranes [13]. The driving forces for fluid movement between the two compartments are the combined differences in diffusive and osmolar flow-driven transport [39, 40]. Creatinine's molecular structure makes it capable of diffusing across the vascular

boundary, but the diffusion is not immediate. A numerical constant describes the rate of diffusion in Table 2.3. Experimental analysis found the concentrations of creatinine in the extravascular compartment to be very similar to SCr measurements [6]. This study also showed when creatinine levels are not static, there is a delayed response in change to creatinine concentration in the interstitial compartment [6].

Figure 2.6 demonstrates the model's ability to achieve equilibrium in all compartments during system homeostasis and nominal kidney function. This is the same setup that was used in Figure 2.3. Since the creatinine concentration model is influenced by the volume model, Fluid bolus influence the concentrations of creatinine within each compartment.

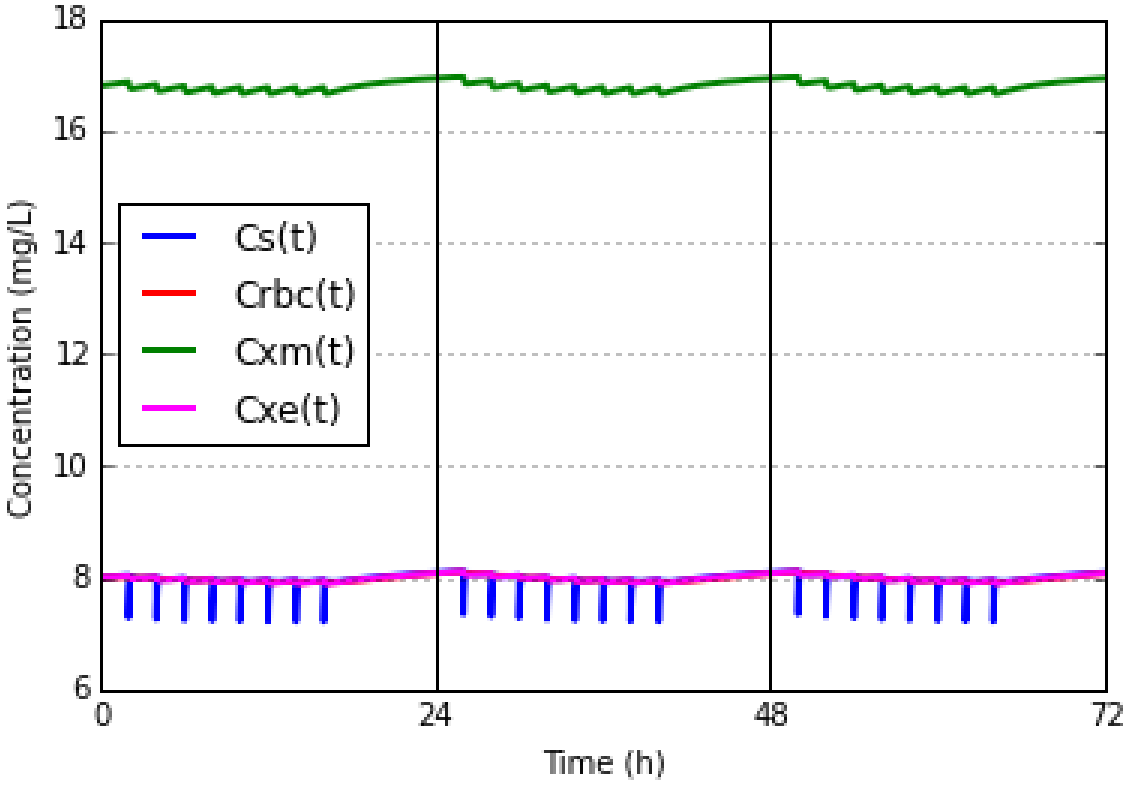

Figure 2.6: Creatinine concentration for each of the four compartments during standard creatinine generation and fluid intake, 100% hydration, and nominal kidney function. Fluid is given to $V_s(t)$ by bolus which simultaneously impacts $C_s(t)$.

Continuous infusion of fluid (2.3 L day^{-1}) for a patient at 100% hydration and nominal kidney function yields constant creatinine concentration in each compartment. Table 2.3 gives the numerical values for all parameters at system homeostasis.

Table 2.3: Creatinine model parameter set assuming a 70 kg simulated patient at volume and creatinine mass equilibrium.

Parameter	Value	Units	Source
Q	5.00	$L \cdot h^{-1}$	[13]
R	2.91	$L \cdot h^{-1}$	[39]
G	44	$mg \cdot h^{-1}$	[44]
k_{gfr}	1.83	h^{-1}	Calculated
Initial Condition	Value	Units	Source
C_s	8	$mg \cdot L^{-1}$	[44]
C_{rbc}	8	$mg \cdot L^{-1}$	[6]
C_{xe}	8	$mg \cdot L^{-1}$	[44]
C_{xm}	12.4	$mg \cdot L^{-1}$	Calculated

2.3 ANALYSIS OF RESULTANT SYSTEM DYNAMICS

In Figures 2.3 and 2.6, fluid was delivered to the model via bolus. In an ICU setting, fluid may be infused intravenously as well as consumed orally. In this section, intravenous fluid delivery is used for hypothetical, clinical scenarios and a 24 hour creatinine analysis.

Here we give 16 hypothetical, clinical scenarios. The 16 patients are evenly distributed to 4 categories: No CKD, stage 2, stage 3, and stage 4 CKD. Each hypothetical patient starts with nominal kidney function and standard fluid intake (2.3 L day^{-1}) for the first day. After day 1, all patients stop fluid intake for 2 days. As a result of no fluid intake, that patients are

taken to the ICU for immediate treatment / care. Each patient is instantaneously given fluid intravenously for a whole day (24 h) at a rate of 0.5 L h^{-1} and their biomarkers are analyzed (t = 72 h). The patients return to the regular fluid infusion rate of 2.3 L day^{-1} proceeding the initial fluid treatment. During fluid resuscitation (t = 80 h), patients 2 through 4 lose varying levels of kidney function. Patient 1 maintains nominal GFR ($GFR_s(t) = 1.0$), patient 2 loses 35% ($GFR_s = 0.65$) of their renal function, patient 3 has kidney function reduced by 70% ($GFR_s(t) = 0.3$), and patient 4 has complete renal failure ($GFR_s(t) = 0.0$). $GFR_s(t)$ remains at this level for the duration of the simulation.

Figure 2.7 shows the total fluid volume of the patient in each of the four scenarios of no CKD. At 100% kidney function, the kidneys are able to maintain system homeostasis as the surplus of fluid given to the model is quickly removed from the entire system. As kidney function decreases for hypothetical patients two through four, we see two notable items occurring. First, when fluid resuscitation is underway, the fluid total in the system increases with decreased kidney function. Secondly, we see that after fluid resuscitation completes and kidney function is decreased, the rate of volume clearance decreases and excess fluid stays in the body for an extended period of time. The top, black, horizontal line identifies 100% hydration for a 70 kg patient. The lower, two lines identify the volumes for moderate and severe dehydration in a 70 kg patient, respectively [40].

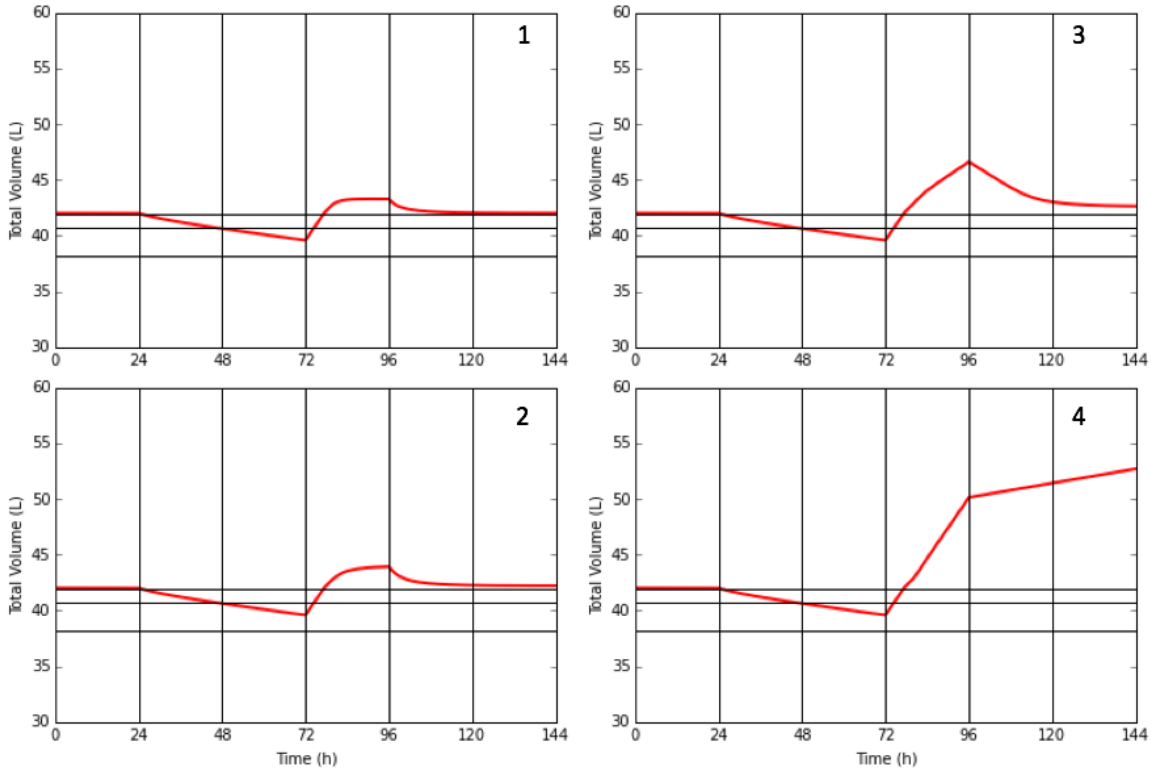

Figure 2.7: Theoretical patients (1-4) resulting volumes at 100% kidney function. Black, horizontal lines represent 100% hydration, minor to moderate dehydration, and severe dehydration in descending order. Patient number is given in the upper, right-hand corner of each subplot.

The impact of systemic fluid trafficking can be observed under the same kidney function scenario where systemic volume management (*i.e.*, urine production rate) is also reduced to 0% of nominal kidney function. Over time the maximum fluid capacity of the vascular compartment (7L) is reached, which then induces interstitial edema, as shown in Figure 2.8. This distribution of fluids is in agreement with physiology [13]. Though a GFR_s of 0 is an extreme case, it identifies the potential hazards of delayed identification of AKI. The model suggests the risk of edema will increase with CKD patients.

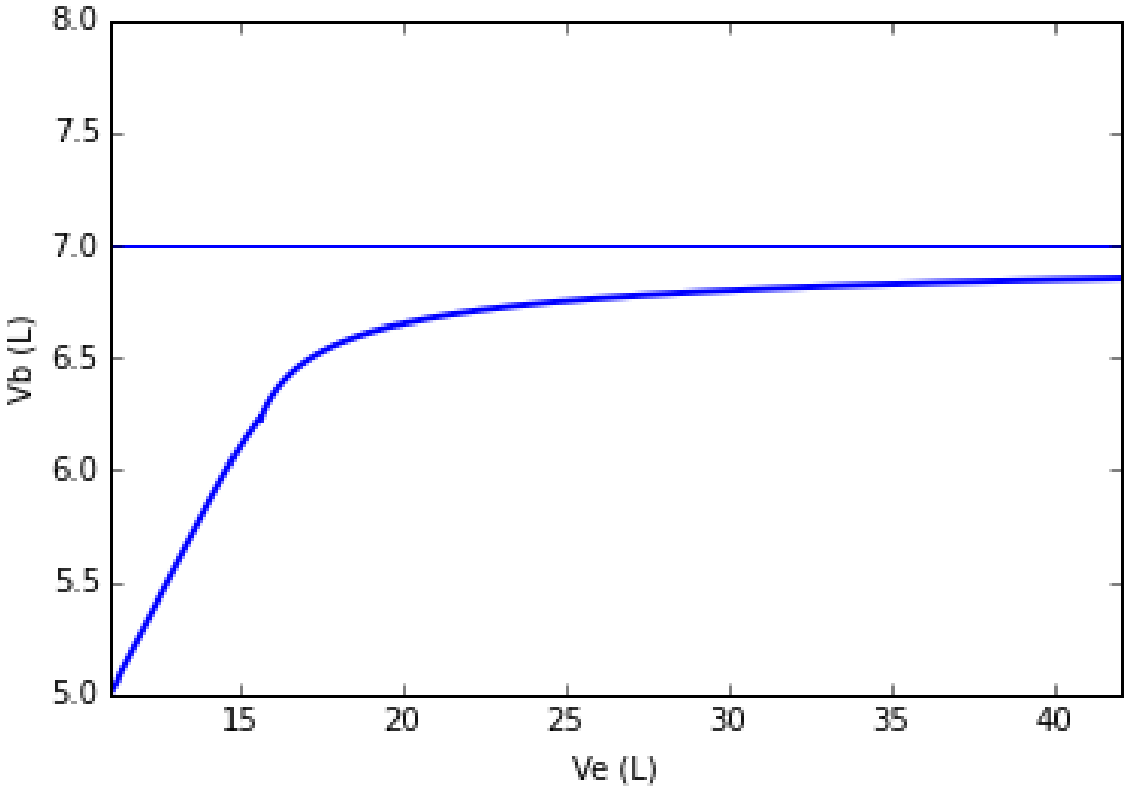

Figure 2.8: A characterization of the vascular volume reaching its maximum capacity and pushing all additional fluid to the extravascular compartment. This illustration maps out the volume distributions within $V_b(t)$ (y-axis) and $V_e(t)$ (x-axis).

Figure 2.9 shows the distribution of fluid throughout the body's four major compartments as the state of edema is approached. It should be noted these distributions will change with different initial volume distribution conditions, volume infusion rates, as well as pre-existing kidney health. The vascular and intracellular compartments reach maximum capacity and additional fluid added to the system is directed to the interstitial compartment. The accumulation of fluid in this compartment will eventually lead to fatal scenarios such as pulmonary edema. Figure 2.9 shows the distribution of fluid in each compartment during the development of interstitial edema. The vascular and intracellular compartments reach maximum volume thresholds and excess fluid is directed to $V_e(t)$.

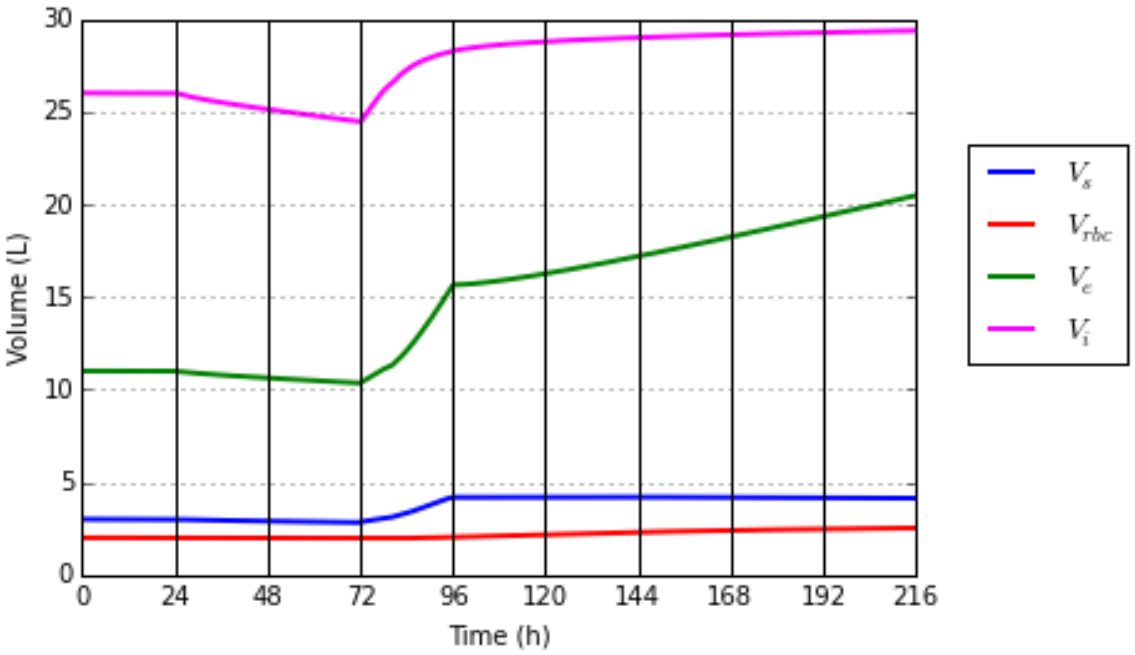

Figure 2.9: Theoretical patient volume distribution during the progression of interstitial edema due to maximum volume compliance of vascular space being reached.

Figure 2.10 shows SCr values for the first four hypothetical patients with no history of CKD. The solid, blue line for each patient identifies SCr for the dynamic model. The dashed line represents SCr for the dynamic model. The loss of kidney function yields an increase in creatinine levels in the vascular, red blood cell, and extravascular compartments ($C_s(t)$, $C_{rbc}(t)$, $C_{xm}(t)$, and $C_{xe}(t)$), respectively. Notable in this figure is the minimal increase in the rate of change in SCr for patient 4 when GFR_s decreases. The rate of creatinine generation accompanied by volume dynamics is responsible for this.

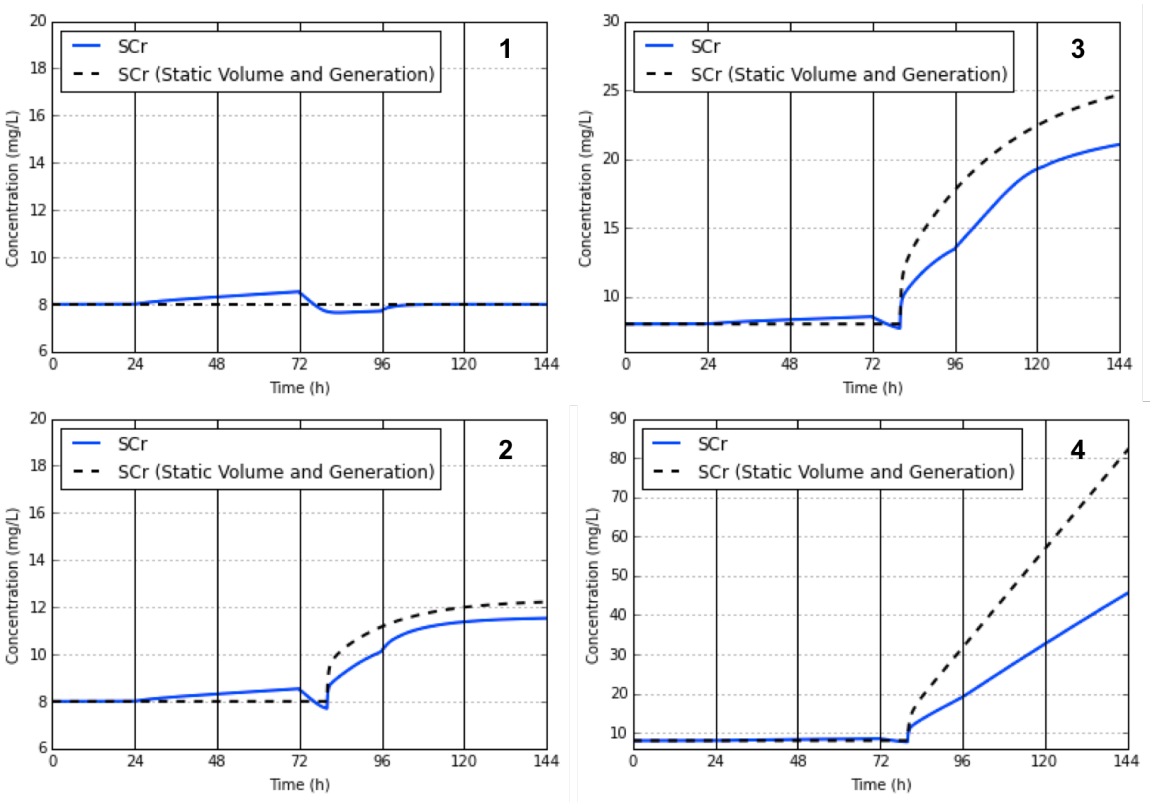

Figure 2.10: Theoretical patients SCr over the course of six days (144h). Baseline SCr is 8 mg L^{-1} (No CKD present). Patient number is given in the upper, right-hand corner of each subplot.

The results in Figure 2.10 are for patients with no CKD. However, the National Institute of Diabetes and Digestive and Kidney Disease (NIDDK) finds that 14% of the United States population suffers from CKD [28]. It has already been shown that an absolute scale is necessary to measure change in SCr for CKD patients [44]. However, the absolute scale has limitations when incorporating volume dynamics and varying creatinine generation rate and should consider these physiological variables. Patients with CKD will hold additional fluid volume within their system and the degree of absolute change in SCr will be amplified as fluid volume varies. This can be seen qualitatively and quantitatively in Figures E2, E4, and 2.12.

The same scenarios are applied to patients with stages 2, 3, and 4 CKD. Baseline SCr values for stages 2, 3, and 4 are given the values 12, 21, and 38 mg L^{-1}, respectively [44]. CKD introduces a new complexity to the model when considering volume dynamics and creatinine generation rate. Since GFR_s influences both of these dynamics, $GFR_s(t)$ will not be equivalent for both the dynamic and static models at identical SCr values. To yield steady states of the dynamic model for any potential baseline SCr value, a library was developed of 1,000 uniformly spaced $GFR_{s,0}$ values from 0 to 1. The code for developing this library is available in Appendix B. This code produced a library of baseline values for both the static scenario as well as the dynamic. Table 2.4 gives the values of $GFR_s(t)$ for all stages of CKD.

Table 2.4: Initial GFR_s Values - Dynamic and Static Model.

Stage of CKD	SCr Baseline (mgL^{-1})	Dynamic Model	Static Model
No CKD	8	1.00	1.00
Stage 2	12	0.62	0.67
Stage 3	21	0.32	0.38
Stage 4	38	0.16	0.21

The library described above was used to determine the initial creatinine concentrations and fluid distribution for all patients with CKD. Briefly, the library considers the initial SCr measurement to be baseline and yields the results for the system to be at equilibrium at that condition. Appendix B shows the process for building this library.

For patients with CKD, analysis of SCr without considering the influence of other dynamics becomes increasingly erroneous. The dynamic model shows how multiple factors should be considered when diagnosing a patient based on SCr measurements and further illustrates the necessity to consider $G(GFR_s(t))$. Figures E1 and 2.12 show the resultant fluid levels and SCr for patients with stage 4 CKD (13 - 16). Results for patients with stages 2 and 3 CKD are in appendix E. Here, we see the difference between the static and dynamic systems.

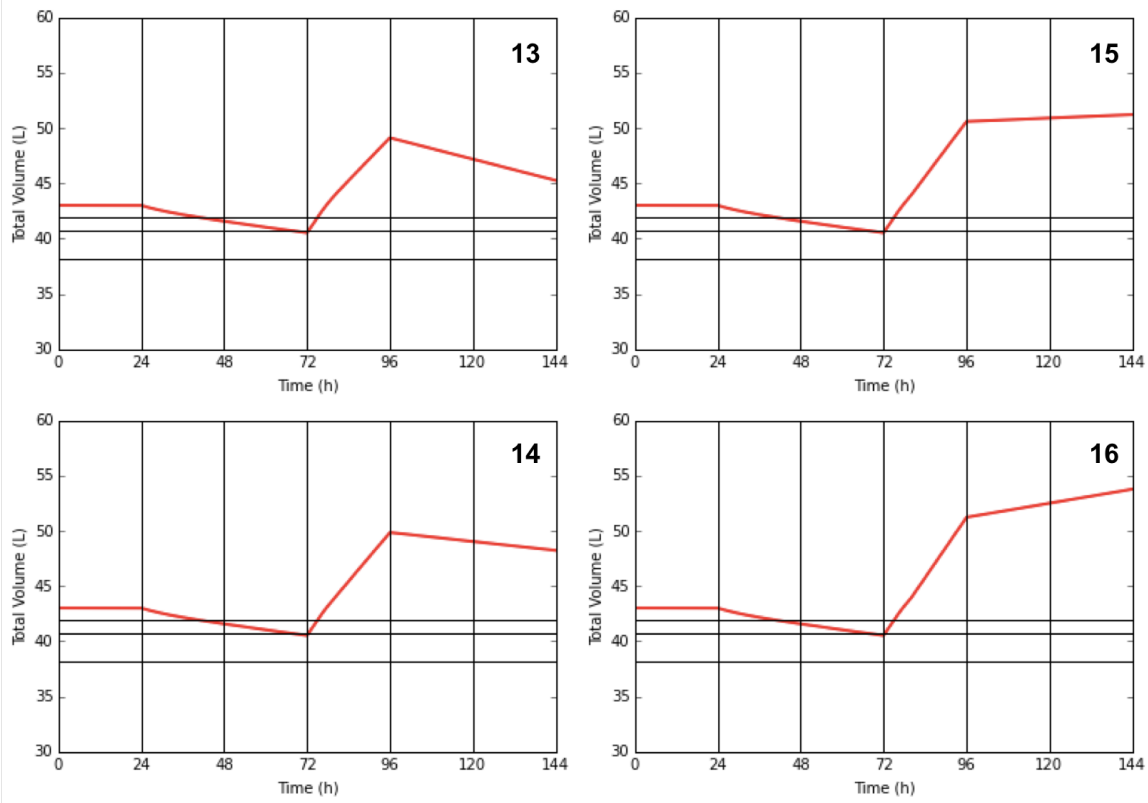

Figure 2.11: Theoretical patients (13-16) resulting volumes when baseline SCr is 38 mg L^{-1}, indicating no potential return to 100% kidney function and nominal k_c. Black, horizontal lines represent 100% hydration, minor dehyrdation, and severe dehydration in descending order. Initial volume distribution set by referencing the patient library for initial conditions to give equilibrium at t=0h. Patient number is given in the upper, right-hand corner of each subplot.

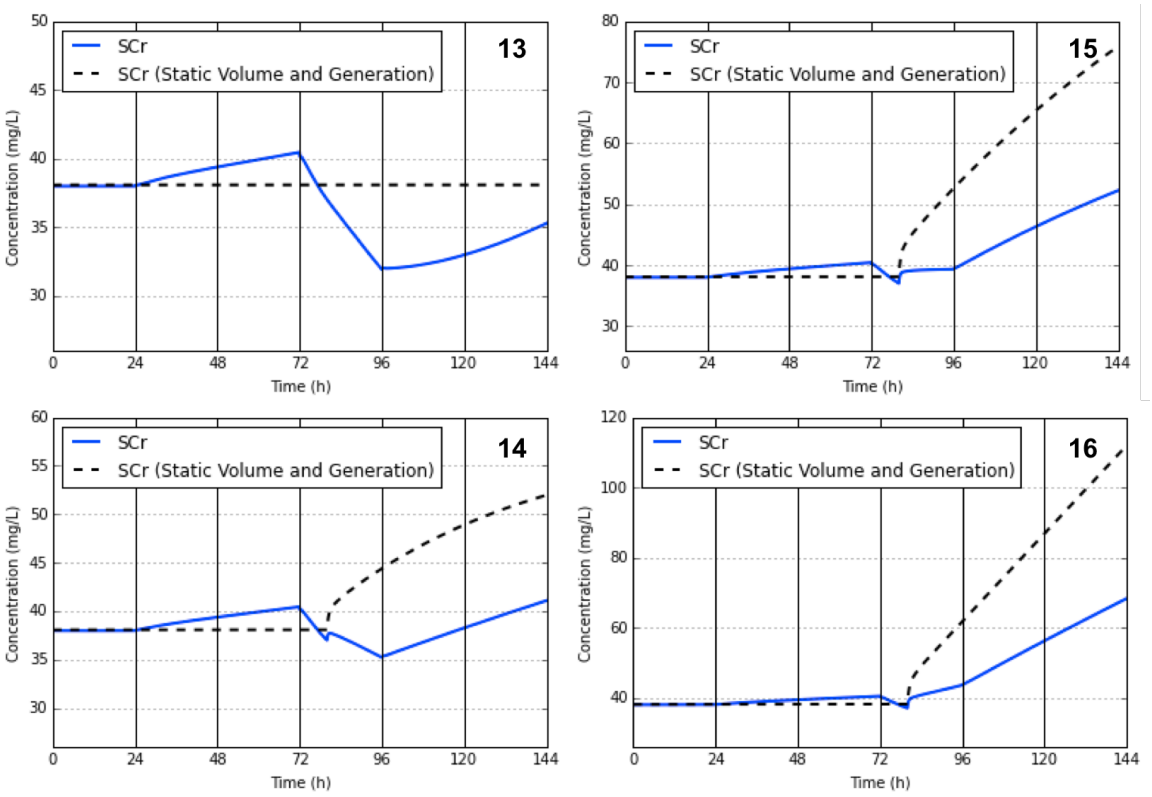

Figure 2.12: Theoretical patients SCr over the course of six days (144h). Baseline SCr is 38 mg L^{-1} (Stage 4 CKD). Patient number is given in the upper, right-hand corner of each subplot.

For hypothetical patients 1 through 16, SCr measurements occur upon theoretical hospital admittance at t = 72h and twenty four hours later at t = 96 h. Table 2.5 identifies the differences in changes predicted by the dynamic and static model at these time points. In patients 1, 5, 9, 10, 13, 14, and 15, the dynamic model shows SCr decreasing between these two time points. The static model fails to detect AKI in these patients the dynamic model successfully predicts. Patients 10, 14, and 15 have a decrease in kidney function by at least 35%. Stage 1 AKI (AKIN definition) is a decrease in GFR by 25% [36]. Patient 15 showed a decrease in SCr by 1.1 mg L^{-1} with 70% reduction in kidney function. For the same patient, The static model predicted a 14.3 mgL^{-1} SCr increase over the same time period. Complete renal failure in a patient with stage 4 CKD, SCr only increased by 3.1 mg L^{-1}. The static model for patient 16 shows an increase in SCr of 23.6 mg L^{-1}. In all cases, the static model predicts too high a change in serum creatinine since it does not account for volume dynamics and creatinine generation rate changes. This limitation of the static model leads to under-prediction of AKI in the ICU. The dynamic model shows an increase in SCr during dehydration. Since the static model does not account for volume dynamics, the increase in SCr due to dehydration would have been a false positive detection of decreased kidney function. The contrast between the dynamic and static models becomes increasingly apparent as the stage of CKD progresses.

Table 2.5: Results for theoretical patients (Dynamic vs. Static models). The total decrease in kidney function at t = 80 h for each patient is given as a % in the column labeled '% Red. GFR_s'.

No CKD, baseline SCr = 8 mgL^{-1}					
Pat. ID	% Red. GFR_s	Dynamic ΔSCr (72 - 96h) mgL^{-1}	Static ΔSCr (72 - 96h) mgL^{-1}	Diff. in ΔSCr (72 - 96h) mgL^{-1}	% Diff. ΔSCr (72 - 96h)
1	0%	-0.8	0.0	0.8	n/a
2	35%	1.6	3.2	1.6	50.3
3	70%	4.9	9.7	4.8	49.6
4	100%	10.5	23.6	13.1	55.6
Stage 2 CKD, baseline SCr = 12 mgL^{-1}					
5	0%	-1.4	0.0	1.4	n/a
6	35%	0.9	4.1	3.2	77.8
7	70%	4.2	11.4	7.1	62.7
8	100%	9.4	23.7	14.3	60.3
Stage 3 CKD, baseline SCr = 21 mgL^{-1}					
9	0%	-4.0	0.0	4.0	n/a
10	35%	-1.3	5.3	6.6	125.2
11	70%	2.5	13.2	10.7	81.3
12	100%	7.1	23.6	16.5	70.0
Stage 4 CKD, baseline SCr = 38 mgL^{-1}					
13	0%	-8.5	0.0	8.5	n/a
14	35%	-5.2	6.3	11.5	182.6
15	70%	-1.1	14.5	15.6	107.8
16	100%	3.1	23.6	20.5	86.9

Volume changes were as great as 27% over 24 hours in the patients listed above. A change this significant should be accounted for within a model interpreting biomarkers collected from the blood plasma. As the degree of renal failure increases, the agreement between the dynamic and static models decrease. Within these 16 scenarios, disagreement in results was found to be as significant as 182.6% (Patient 14) with a decrease in kidney function of just 35%. These results show the importance of considering the volume and creatinine dynamics presented in this work.

Blood tests in the ICU occur approximately once every 24 hours [10, 11]. SCr may increase, decrease, or stay constant as the baseline GFR_s value changes. However, kidney function may have not remained constant. Figure 2.13 demonstrates this and shows why one method of detection of AKI may not be enough to determine the true state of an ICU patient. This figure compares the results when considering creatinine generation rate and volume dynamics versus static fluid volume and no change in creatinine generation rate. For all cases, at the first time point, the hydration is at equilibrium and fluid is continually infused to yield 2.3 L day^{-1}.

The model is built using a baseline SCr value of 8 mg L^{-1}. The points on the chart identify the initial value of $GFR_s(t)$ and the amount of time $GFR_s(t)$ stays at this value. When t is greater than the time designated by the point, $GFR_s(t)$ returns to nominal conditions. Throughout the process, the rate of fluid consumption is constant at 2.3 L h^{-1}.

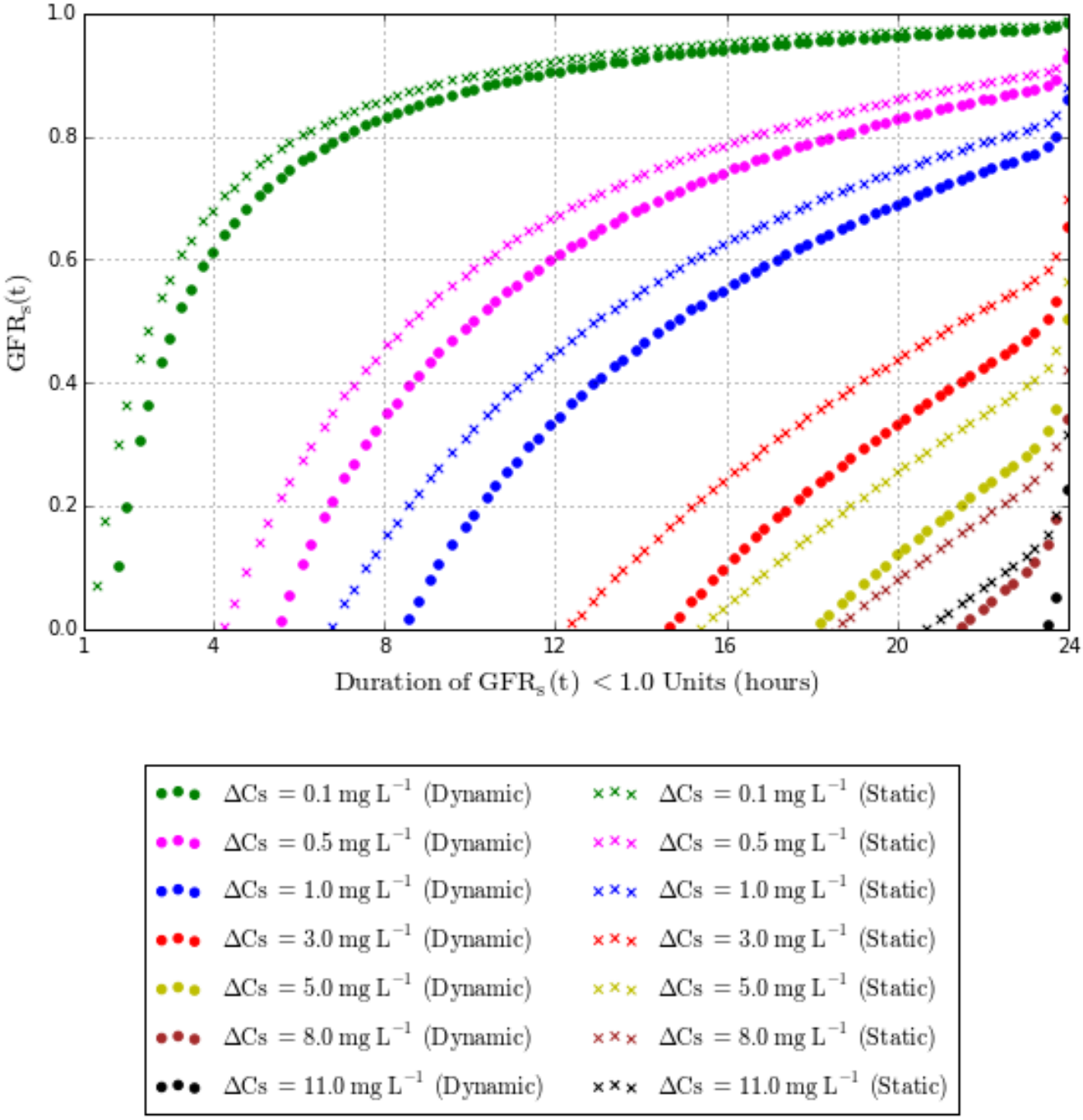

Figure 2.13: Possible $GFR_s(t)$ scenarios to yield the observed ΔSCr in a 24h window. Solid circles represent $GFR_s(t)$ results for the dynamic model. Points labeled with "x" represent $GFR_s(t)$ with a static model. Initial SCr measurement is 8 mg L^{-1}.

Figure (2.13) highlights that a static model will under-predict the depth of kidney damage and/or the duration of kidney injury for a given ΔSCr. This becomes increasingly apparent as ΔSCr increases from 0.1 to 11 mg L^{-1}. Figure (2.13) also shows that a static model predicts too large of a range of potential $GFR_s(t)$ outcomes that a true, physiological environment would not permit. Both models at lower ΔSCr values have a large range of outcomes for $GFR_s(t)$. An increased rate of biomarker analysis would decrease the range of outcomes. Urine output rate monitoring over the 24 hour window would also enhance the prediction and/or duration of kidney injury for a patient.

3.0 MODEL FITTING TO PATIENT DATA

This chapter examines the ability of the model to fit actual patient data provided by UPMC. First, we describe criteria for selecting the patients to be analyzed. Pairing the data with the model will require processing the data, setting initial conditions, and developing rules and constraints. These items are discussed in section 3.1.4.

3.1 PATIENT SELECTION AND MODEL FITTING PROCESS

3.1.1 Motivation for Model Fitting

Up to this point, we have discussed the theoretical development and advantages of this model. The intent of this chapter is to show the performance of the model with real patient data over an array of ΔSCr, initial conditions, volume dynamics, and final patient outcomes. The previous chapter showed a dynamic model can provide a more timely diagnosis of AKI. The goal of this chapter is to demonstrate that the model can capture actual patient data from a clinical care cohort. We follow criteria outlined in Section 3.1.2 to identify ten patients from the UPMC database that we want to be able to capture.

3.1.2 Patient Data Selection Process

All ten patients stayed in the hospital for a range of ten to twenty days. Since the model was developed for a theoretical patient weighing 70 kg, all patients weighed between 120 and 180 pounds. None of these patients were on medical devices other then catheters. The first measurement of SCr was required to be greater than 0 mg L^{-1} and less than 30 mg

L^{-1}. If the first time point was not the minimum value of SCr recorded, an extra time point was added to the beginning of the timeframe and given the minimum SCr value for that patient. This allows the model to set a baseline SCr value at t = 0 h. Within the given timeframe, patients 1 - 5 survived and patients 6 - 10 did not. The outcome of patients after their hospital stay was not considered. Both surviving patients and non-surviving were given their own categories. The two categories were made; each contained four subcategories. The first subcategory was comprised of patients having SCr measurements that changed by 10 to 30 mg L^{-1}. The second subcategory consisted of patients with SCr values that changed by 30 to 50mgL^{-1}, compartment three by 50 to 70 mg L^{-1} and the fourth compartment by greater than 70 mg L^{-1}. One patient would be randomly selected for each subcategory except for subcategory four. This subcategory contained two patients. The selection criteria had no preference to the patient's baseline SCr value. Figure (3.1) gives the distribution of the 10 patients.

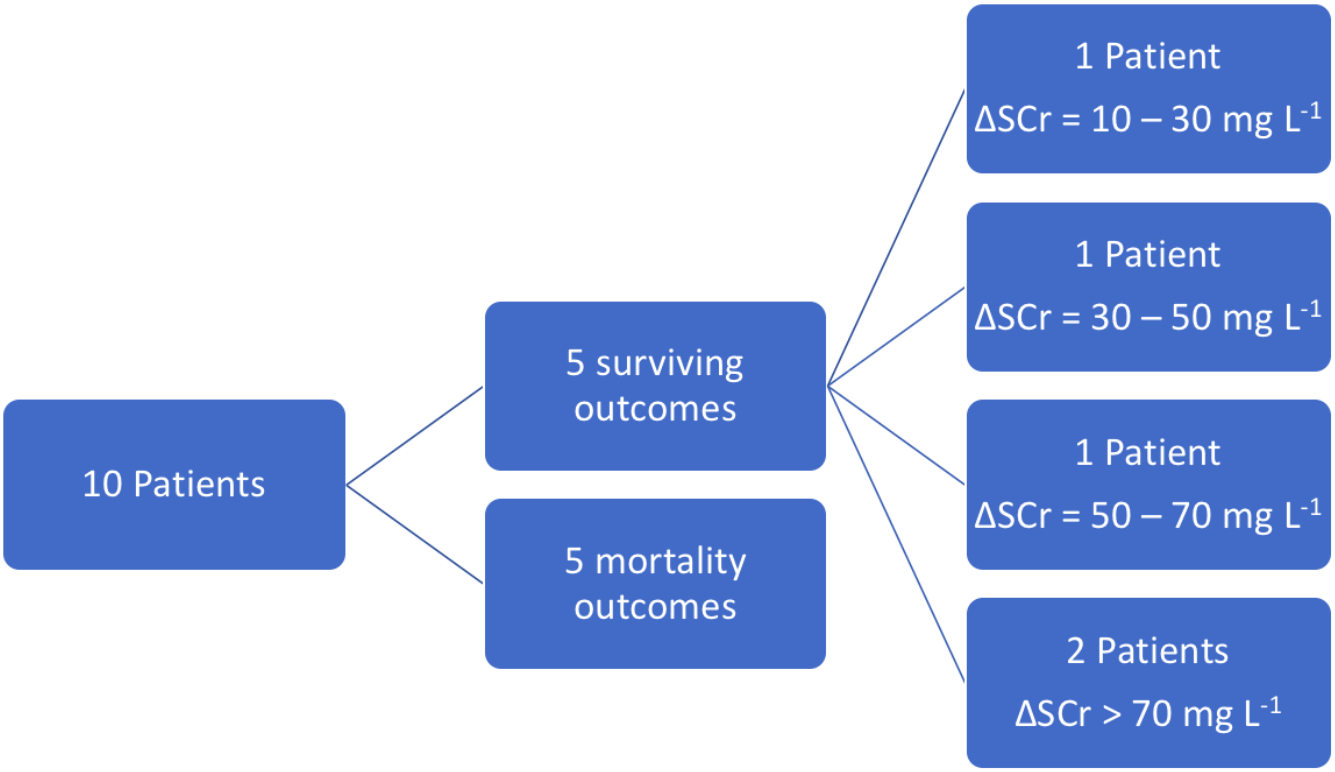

Figure 3.1: UPMC patient category distribution. Subcategories are given for the surviving patients. The same subcategories are used for mortality patients.

3.1.3 Tools Utilized

For patient data processing, optimization, and graphing the following programs for Python v3.4 were utilized: pyomo [15], pandas [26], numpy [42], and matplotlib [18]. For patient data filtration and selection, MySQL Workbench was used along with numpy, mysql.connector, and datetime from Python v3.4.

3.1.4 Data Processing, Constraints, and Other Critical Factors for Optimization

Data of interest for each patient was fluid administered or consumed, fluid exiting via the urinary tract (k_c) or interstitial compartment, and measured serum creatinine values.

3.1.5 Fluid Input

There are seven different methods of fluid administration among the ten patients. These methods include oral intake, intravenous therapy (IV), continuous infusion, feeding tube intake, blood products/colloids, operating room intake, enteral intake. Oral intake is the delivery of fluid by way of oral consumption. IV is given by direct injection to the vascular compartment over short periods of time for reasons such as drug delivery. Continuous infusion is also given directly to the blood and is the controlled administration of fluid over an extended period of time. Blood products are components of blood such as red blood cells, platelets, fresh frozen plasma, and cryoprecipitate [20]. Since the blood product isn't specific whether it is red blood cells or plasma, the assumption is made that all blood product components are directed to the plasma. Colloids refer to plasma substitutes [20]. Operating room intake is fluid input during operations. Enteral intake is nutrition delivered by way of a feeding tube.

The medical record indicates times ($t_{n,\ fluid}$) and total volumes of n individual fluid inputs. The fluid is cumulatively administered from $t_{n-1,\ fluid}$ to $t_{n,\ fluid}$ at a constant rate that sums to the total volume given. This method of fluid input can be described as a rectangular wave. The first data point in all data sets is a SCr measurement and marks the start ($t = 0$ h) of the patient's clinical visit. Some patients have fluid given at the initial time point. If the medical record indicates fluid given at $t = 0$ h, it is administered to the patient as $V_s(0) = 3L+B_s$ where B_s is the indicated volume.

Each type of fluid was individually directed to the plasma compartment. In reality the dynamics of the different methods of fluid input may differ. Here, all fluid inputs follow the same set of rules.

Interstitial fluid loss (E) leaves by way of a rectangular wave as well. Its method of leaving the model follows the same rules as fluid input.

3.1.6 Initial Conditions and Constraints

The initial conditions came from both the given data and model assumptions. Baseline SCr values were given by compartment C_s at $t = 0$ h. If SCr_0 was 8 mg L^{-1} (indicating nominal

kidney function), the total volume of the model ($V_{tot}(t)$) is given by the patient library in Appendix B +/- any given fluid administered and/or exiting at t= 0 h. If a patient has a baseline SCr measurement of 8 mg L^{-1}, then at t = 0 h, $V_i(0)$ = 26 L, $V_e(0)$ = 11 L - B_e, $V_s(0)$ = 3 L + B_s, and $V_{rbc}(0)$ = 2 L. These initial volumes vary with baseline SCr.

$GFR_s(t)$ is set to predict a single value of kidney function between adjacent SCr measurements. The range of possible $GFR_s(t)$ values is set by baseline SCr. $GFR_s(t)$ can take any value from 0 to 8 mg L^{-1} / SCr_0 where $GFR_{s,max} = 1$ represents nominal kidney function and no CKD. The reason for the limitation on the range of $GFR_{s,max}$ values is patients with CKD will not have a fully functional kidney before or after AKI. Therefore, SCr_0 dictates $GFR_{s,max}$. A patient could enter the hospital with AKI, but this model assumes the minimum value to be the level of CKD. To further limit the maximum change in $GFR_s(t)$, $GFR_s(t)$ is allowed to vary by 0.5 units from one prediction to the next. This rule is not physiologically fortified, but is set to further constrain $GFR_s(t)$.

The initial level of hydration for each patient is unknown when they enter the ICU. Since urine production rate can be influenced by hydration level and kidney function ($GFR_s(t)$), the equation for k_c requires the ability to adapt per patient. For example, if a patient was significantly dehydrated and experiencing severe changes in $GFR_s(t)$, according to Equation (2.9), we would expect minimal influence on urine production rate from kidney function. Equation (3.1) gives the form of k_c allowing for flexibility in the expression from one patient to the next.

$$k_c(GFR_s(t), V_{tot}(t)) = k_{c,min} + GFR_s \frac{k_{c,max} - k_{c,min}}{1 + e^{k_{c,slope}(V_{tot}(t) - k_{c,vol})}} \tag{3.1}$$

Likewise, Equation (2.9) says if the patient was initially in a state of edema, we would expect significant correlation of change in $GFR_s(t)$ being equivalent to change in k_c. Since the hydration level of each patient is unknown, the model is given the ability to shift the location of the slope connecting $k_{c,min}$ and $k_{c,max}$. In addition, the maximum and minimum rates of of urine production per patient are subject to deviation from literature values and are given flexibility. As is seen in the patients later in this chapter, these rates of urine

production can vary. Table 3.1 captures the upper bound, lower bound, and initial guess for each of these parameters that make up k_c.

Table 3.1: k_c parameter ranges and initial guesses

Parameter	Range	Initial Guess
$k_{c,min}$	[0.001, 0.020]	0.009
$k_{c,max}$	[0.6, 1.2]	0.833
$k_{c,slope}$	[-3.0, -1.5]	-2.25
$k_{c,vol}$	[25.2, 55.2]	38.0

The objective function of SCr and urine output data is minimized: The squared error of urine output total (UOT) recorded over time versus the model's prediction of total urine production, the squared error of recorded SCr over time versus the model's prediction, change in $GFR_s(t)$ from one time point (in the overall index) to the next, change in urine production rate from one time point (in the overall index) to the next, and change in SCr from one time point (in the overall index) to the next. The highest priority is given to minimizing the urine production rate and SCr values. Limiting the degrees of freedom of $GFR_s(t)$ was important for the stability of the fitting algorithm. To evaluate the model's fit to the data of each patient, SSE, MSE and RMSE are calculated. To limit the degrees of freedom, the objective function penalizes change in SCr, k_c, and $GFR_s(t)$ by Equation (3.2) (when $n > 0$). Coefficients prior to the mathematical expressions dictate the priority of an attribute in the objective function.

$$SD = \sum_{n=0}^{n,max-1} \left(\frac{f(x_n) - f(x_{n-1})}{t_n - t_{n-1}} \right)^2, n > 0 \tag{3.2}$$

The solver used for the optimization process is "Interior Point OPTimizer" (IPOPT) [43]. IPOPT is designed for large-scale optimization by finding local solutions to continuous systems [43]. The maximum number of iterations given for the process to reach an optimal

solution is $1x10^4$ and the tolerance is set to $1x10^{-6}$. The tolerance is the threshold of acceptable change in two consecutive objective function scores and defines the end-point to the optimization process.

3.2 RESULTS AND DISCUSSION

The results for the ten patients is presented in Figures 3.2 through 3.11. Patients 1 through 5 survived their stay in the hospital and patients 6 through 10 did not. Each of these figures consist of nine subplots and are labeled 'a' through 'i' in alphabetical order. 'a' details fluid volume distribution in the patient, 'b' indicates the rate of volume input into the model (all fluid enters the model by way of the plasma compartment via rectangular waves), 'c' gives the total UOT in the model over time, 'd' gives fluid distribution only within the blood compartments to detail what the impact is on SCr, 'e' gives the fluid volume in the interstitial compartment and the rate of interstitial fluid leaving the model, 'f' gives the total urine output over time recorded by hospital employees and the fitted total urine output given by k_c, 'g' is the resultant $GFR_s(t)$ based on the fitting of SCr and k_c, 'h' details k_c and is the derivative of subplot 'f' with respect to time, and 'i' is the fitting of SCr predicted by the model to values from the medical record.

3.2.1 Surviving Patient Fits

In Figure 3.2, patient 1 maintained fairly stable, nominal hydration levels throughout the 309 hour hospital stay as shown by $k_{c,vol}$ in Table 3.3. At this state of hydration, we expect to see changes in $GFR_s(t)$ impacting both the rate of urine output as well as SCr. Of note, the model decreases in k_c when SCr increases at t=75h and an increase in urine output rate directly proceeding SCr recovery at t=100h. Three different types of fluid were administered to this patient including IV, oral intake, and continual infusion. The parameters for k_c are given in Table 3.2. No fluid exited the system by way of the interstitial compartment. SCr has a minimum value of 8 mg L^{-1}, meaning the system is capable of 100% kidney

recovery ($GFR_s(t)$=1). The maximum SCr measurement for patient 1 was 20mgL^{-1}, yielding a maximum ΔSCr of 12 mg L^{-1}. $GFR_s(t)$ initiated at 20% nominal kidney function. Table 3.2 gives the SSE, MSE, and RMSE of the urine and SCr fits. Overall, the model was able to replicate patient 1 except for the end results due to the optimization formulation. At approximately t = 260 h, the urine output rate significantly increased and the model did not respond to it. The SCr RMSE for patient 1 (1.32 mg L^{-1}) is less then the average SCr RMSE for the 5 surviving patients (1.82 mg L^{-1}). UOT RMSE for patient 1 (0.81 L) is also less then the average UOT RMSE (1.10 L) for the 5 surviving patients. The RMSE values show the approximate fit for the model is fairly close to the actual actual data.

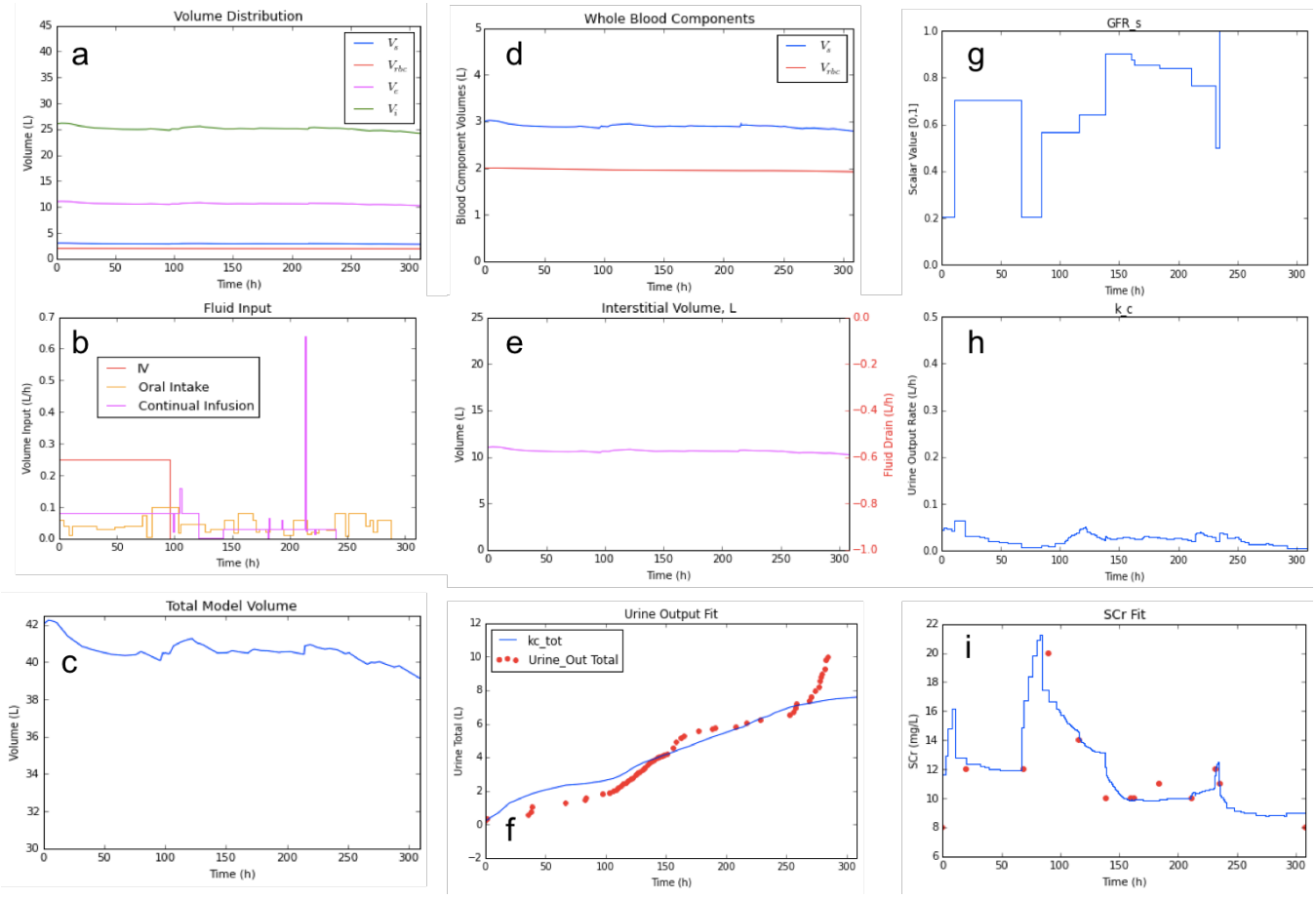

Figure 3.2: Patient 1 (surviving) data and fitted results. Time in hospital = 309 h.

Patient 2 plots are given in Figure 3.3. According to the model output, fluid levels were low for the patient upon arrival as $k_{c,vol}$ was at the maximum value of 55.20 L. The

minimum and maximum values of the $k_{c,vol}$ window were set to allow for a large range of initial values. At the state of dehydration predicted by $k_{c,vol}$, we expect to see changes in SCr and GFR_s not influencing k_c and the following results occurred. In subplot 'f', at t=25h, SCr begins to increase as the urine output rate stays relatively constant. The fluid types given to patient 2 were IV, oral intake, continuous infusion, and blood products / colloids. The rate of IV intake at roughly t=240h is abnormally high, but only for a brief moment. The reason for this significant increase is unknown. Baseline SCr for this patient was 24 mgL^{-1}, meaning the maximum value of GFR_s is 0.33. The maximum SCr measurement for patient 2 was $35mgL^{-1}$, yielding a maximum ΔSCr of 11 mgL^{-1}. As fluids continue to be infused into patient 2, the hydration level approaches the value given by $k_{c,vol}$ and the rate of urine output then significantly increases. As stated before, the initial volume of 42 L for the model is not indicative of 100% hydration. The hydration level of a patient entering the ICU is unknown and therefore the model must adjust for this uncertainty. The parameter $k_{c,vol}$ is set to predict the fluid level of the patient at ICU admittance. If $k_{c,vol}$ is greater than 43.18L (referred to Equation (2.9)), it is indicative of a dehydrated patient. Similarly, if it is less than 43.18L, the patient is considered to be fully hydrated. Subplot 'c' shows the gradual increase of total volume as 100% hydration is achieved at approximately t=230h. For this patient, no fluid leaves the model by way of the interstitial compartment. The constraints given to $GFR_s(t)$ may not have been significant enough as subplots 'g' and 'i' oscillate more than what would be desired. SCr RMSE for patient 2 (0.11 mg L^{-1}) is also abnormally low compared to the other patients. The oscillatory motion of $GFR_s(t)$ for this patient is not ideal. Steady increases and decreases of $GFR_s(t)$ over time would be more in line with actual changes in kidney function. UOT RMSE was typical of other values for living patients. If a method could be developed to increase the penalty to $GFR_s(t)$, the fit would be more more in line with reality.

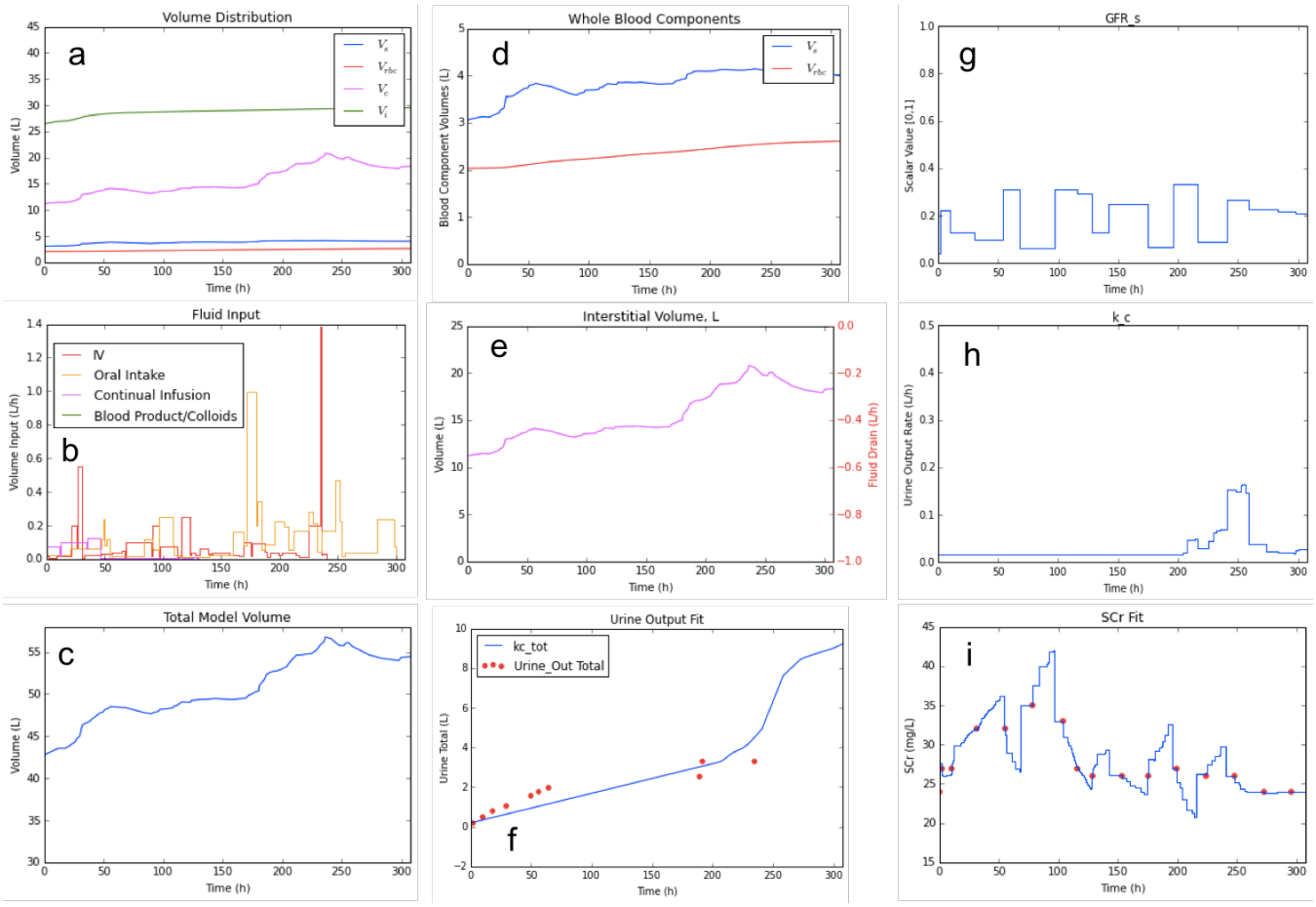

Figure 3.3: Patient 2 (surviving) data and fitted results. Time in hospital = 307 h.

In Figure 3.4, subplots for patient 3 are given. Similar to patient 2, the value of $k_{c,vol}$ indicates the patient's fluid levels were low upon arrival to the hospital. Both, the data and the predicted urine output total in subplot 'f' verify this line of thought. The rate of urine production stays constant as $GFR_s(t)$ varies with SCr. Upon admittance to the hospital, the patient's SCr value increased for 108 hours from 13 mg L^{-1} to 76 mgL^{-1} for a total change of 63 mgL^{-1}. Past 108 hours, SCr data and predicted values begin to oscillate as does GFR_s with no change to the rate of k_c. Since the minimum value reported for SCr is $13mgL^{-1}$, the maximum achievable value of GFR_s is 0.62 using the equation previously described. Five types of fluid delivery were used on patient 3: IV, oral intake, continual infusion, blood products / colloids,

and operation intake. When the patient initially entered the hospital, they received a large amount of fluid by way of operational intake and IV. This increased $V_{tot}(t)$ to near 100% hydration, however, the significant fluid losses had an impact on the total volume as seen in subplot 'e'. Significant loss of fluid in 'e' occurs at approximately t=140h and its impact can be seen on the components of hematocrit measurements in subplot 'd'. The blood serum decreased by approximately 25% as fluids shift to achieve system homeostasis. During the patient's stay in the hospital, SCr gradually decreases, but never fully recovers to baseline. SCr and UOT RMSE values for patient 3 suggest a good fit to the provided data. Patient 3 highlights that increased variance in SCr with limited change change in k_c is an indicator of dehydration.

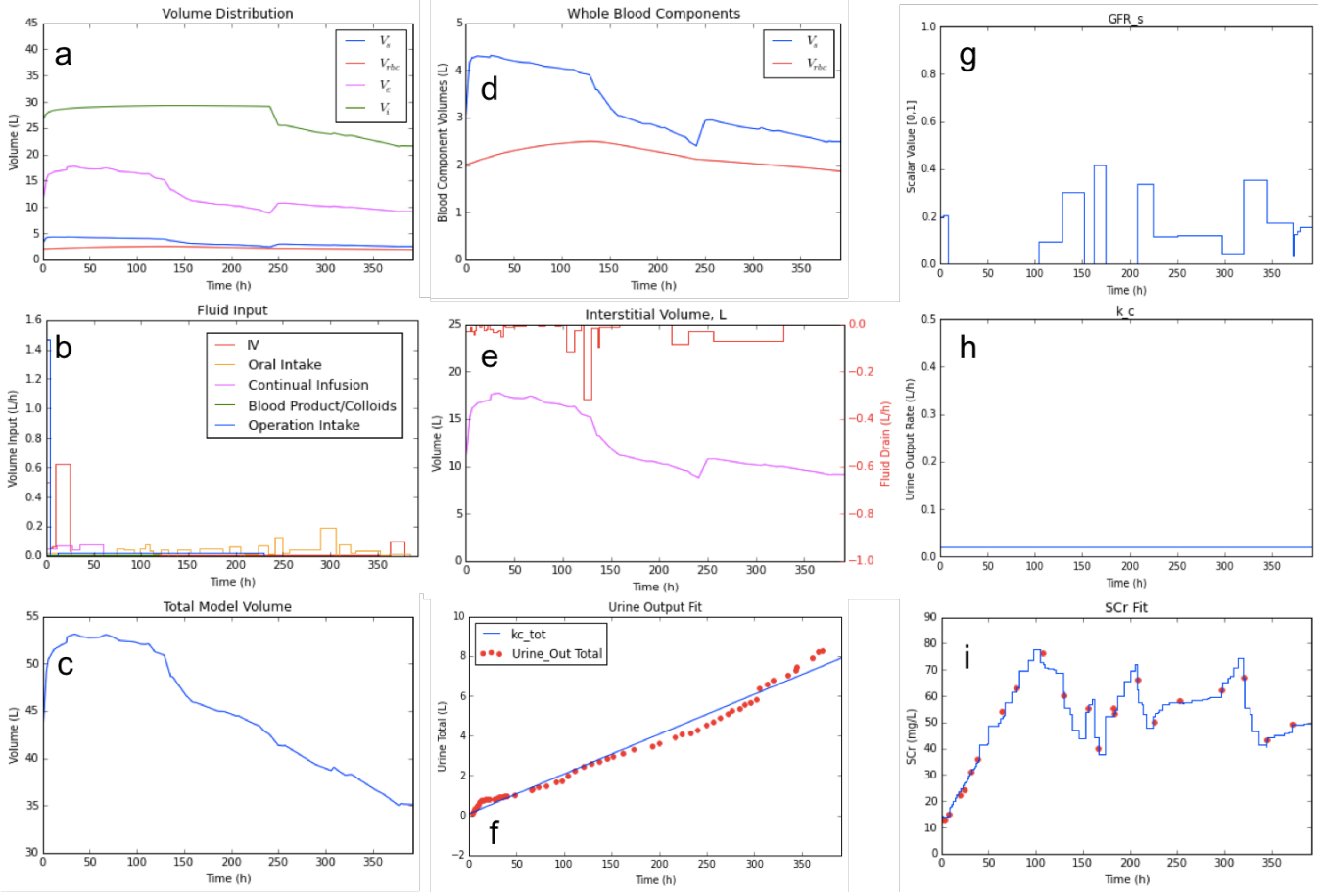

Figure 3.4: Patient 3 (surviving) data and fitted results. Time in hospital = 392 h.

Patient 4 optimization results are provided in Figure 3.5. This patient's $k_{c,vol}$ value is the opposite of patients 2 and 3. Here, the value is 25.8L indicating the patient's initial state of hydration being overly hydrated. Subplot 'f' hints that $k_{c,vol}$ could have been lower as the urine production rate predicted by the model drops off toward the end of the analysis as the data for urine production total suggests the uring production rate stays constant throughout the patient's stay in the hospital. A value below 42 L for $k_{c,vol}$ suggests a correlation of SCr with k_c as $GFR_s(t)$ varies over time. Here, we see an increase in SCr from 7 mgL^{-1} to 95 mgL^{-1}. A baseline value of $\leq$8 mgL^{-1} permits GFR_s to choose any value between 0 and 1. The model predicts GFR_s to be 0 even upon admittance to the hospital with an SCr measurement below 8 mg L^{-1}. At the maximum measurement of SCr at roughly t=200h, GFR_s improved from 0 to 0.2. $GFR_s(t)$ slowly improves throughout the patient's stay in the hospital as subplot 'g' shows. As the health of patient 4 improves, k_c significantly increases, releasing fluid until the system reaches homeostasis as shown in subplot 'h'. This patient received fluids by way of 5 different methods: IV, oral intake, continuous infusion, blood products / colloids, and a feeding tube. In subplot 'b', we see a significant, brief spike in fluid infusion via feeding tube at approximately t=140h. This event appears to be correlated with the significant loss in interstitial fluid that occurs between the time points of 75h and 130h. Patient 4 continually lost fluid via the interstitial compartment throughout their stay in the hospital. Patient 4 undergoes a significant change in SCr and k_c, yet the model performs effectively to mimic the data collected for both urine production and SCr. This yielded a good understanding of the kidney's health via $GFR_s(t)$. The poor score for SCr RMSE can be attributed to the model's inability to capture the SCr measurement at t = 250 h. Notice also at t = 250 h, the rate of urine production decreases. Subplot 'c' shows a significant decrease in $V_{tot}(t)$ partly due to interstitial fluid loss. The model was not capable of fully capturing the volume of patient 4 past t = 250 h and this played into the poor RMSE scores. From a qualitative viewpoint, the model fit the data well.

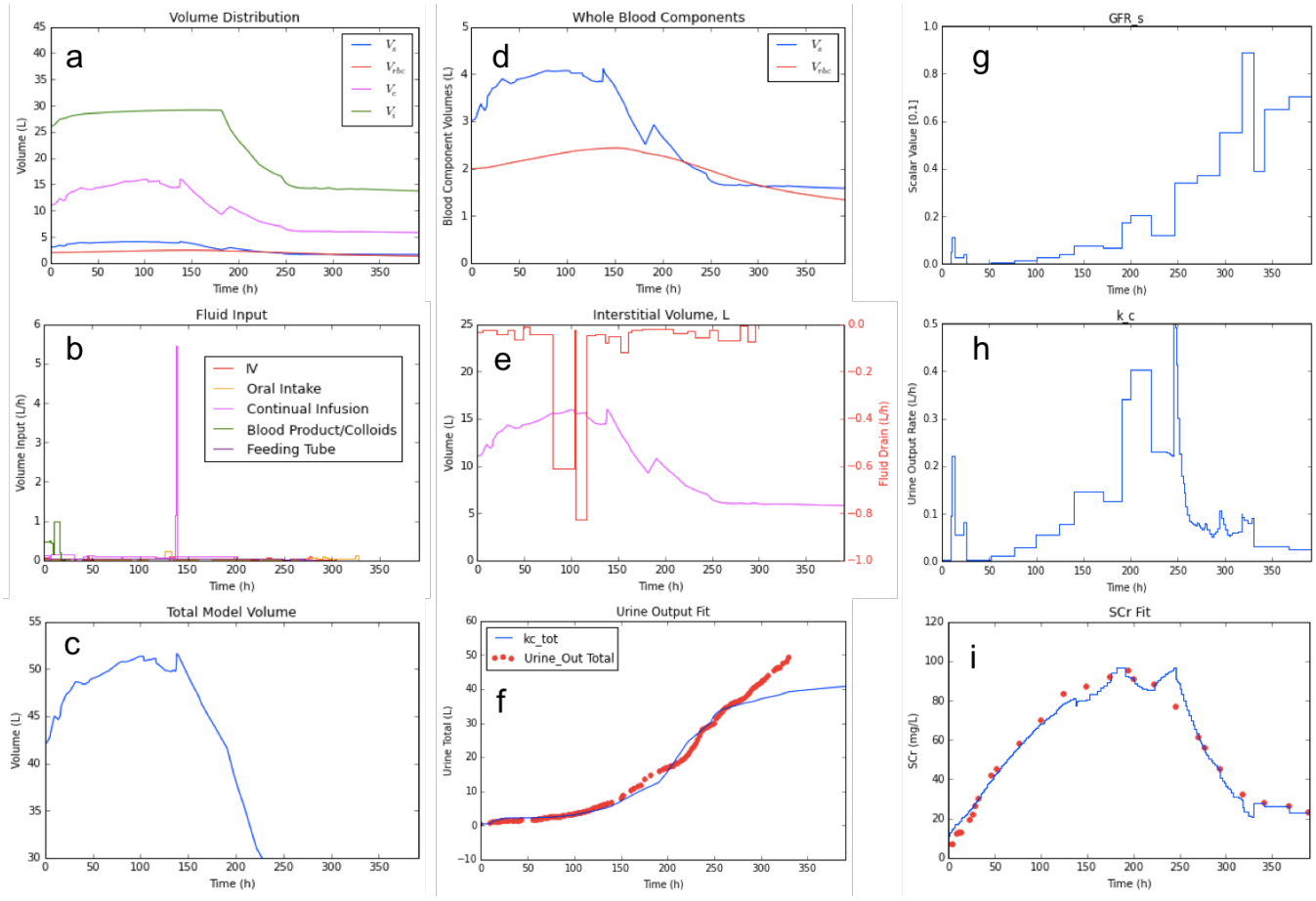

Figure 3.5: Patient 4 (surviving) data and fitted results. Time in hospital = 391 h.

Figure 3.6 shows the results of the 5th patient's optimization. Over the course of 342h, three types of fluid were administered: IV, oral intake, and continual infusion. For the first four days, the patient was losing fluid from the interstitial compartment at a rate of 0.1 L h^{-1}. During the same window of time, there is an increase in the rate of IV input and continuous infusion. According to the model, the patient was at approximately 100% hydration upon their admittance to the hospital. As $GFR_s(t)$ changed, changes to SCr and k_c responded accordingly and fit the given data. Likewise, the rate of urine output corresponded to changes in GFR_s and these changes fit the data. Patient 5 had a baseline SCr value of 19 mg L^{-1}, meaning $GFR_{s,max}$=0.42. The total change in SCr was 73 mg L^{-1} and was the second largest

increase of the five surviving patients. RMSE for both k_c and SCr in Table 3.2 indicate a good fit to the data and a qualitative analysis would agree with that conclusion.

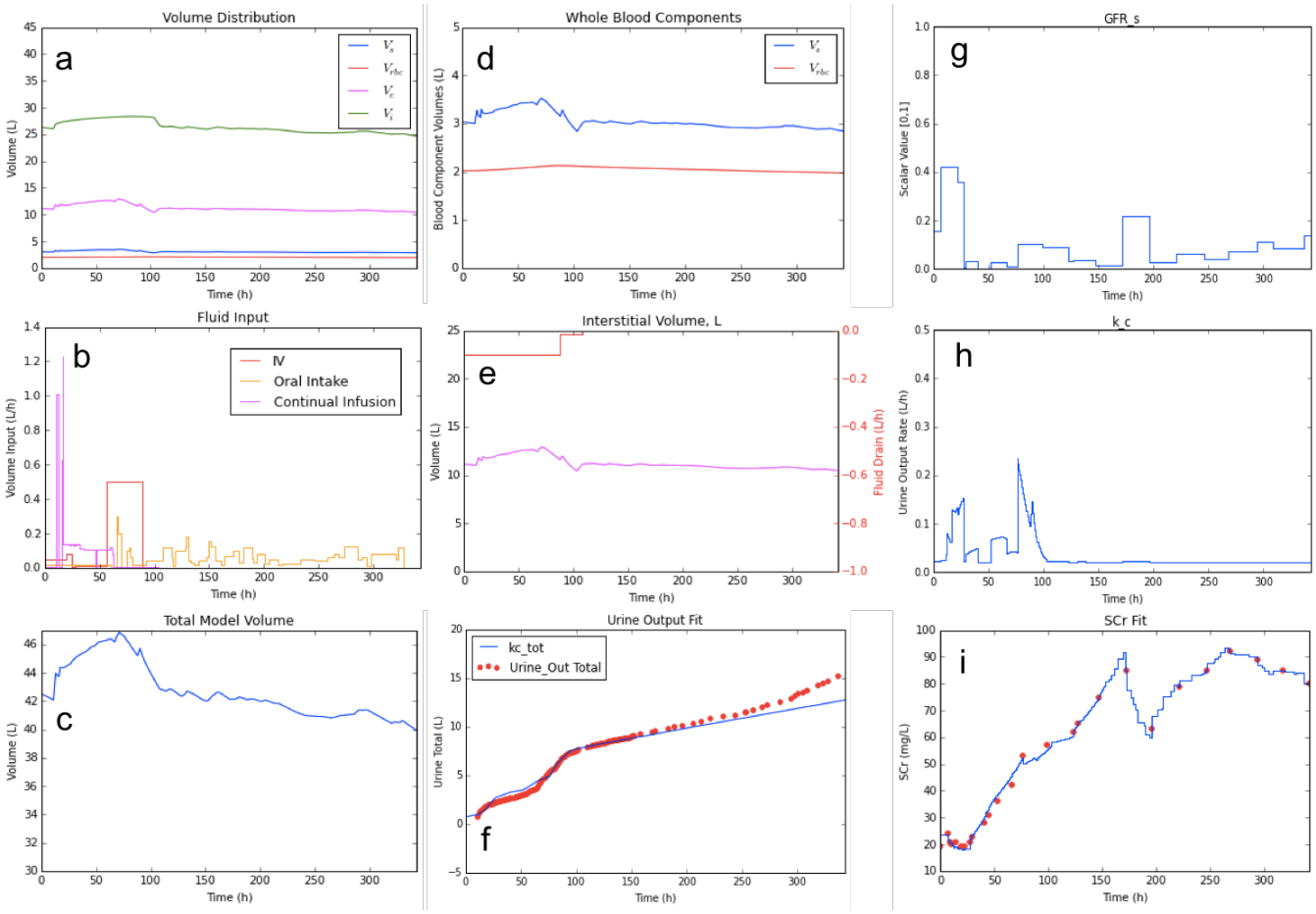

Figure 3.6: Patient 5 (surviving) data and fitted results. Time in hospital = 342 h.

3.2.2 Mortality Outcome Patients

Patient 6 (Figure 3.7) did not survive their stay in the hospital. Over the course of 399 hours, patient 6 received fluids by IV, oral intake, and continual infusion. $k_{c,vol}$ indicates a potentially low total fluid volume throughout their stay. From t = 140 h to t = 325 h, SCr increased by 1.8 mg L^{-1} day^{-1}. During this time period, there was minimal change in urine production rate. This complements the model's assessment of low fluid levels. According to subplot 'b', the rate of continual infusion of fluids for patient 6 was 0.075 L h^{-1} which is

less then the rate of fluid intake required for the model to maintain equilibrium. Fluid levels slowly decrease throughout the patient's 16 days. Subplot 'e' shows no loss of fluid by way of $V_e(t)$. Studying all 'b' subplots, the collective rate of fluid consumption for patient 6 is significantly less then any of the other 10 patients. ΔSCr for this patient was 12 mg L^{-1}. Subplot 'g' shows a gradual decrease in kidney function over time - from 55% to roughly 35%. SCr and UOT RMSE values indicate a good fit and is comparable to the results of the surviving patients. Minimal oscillatory motion is seen with $GFR_s(t)$ which further indicates a good fit.

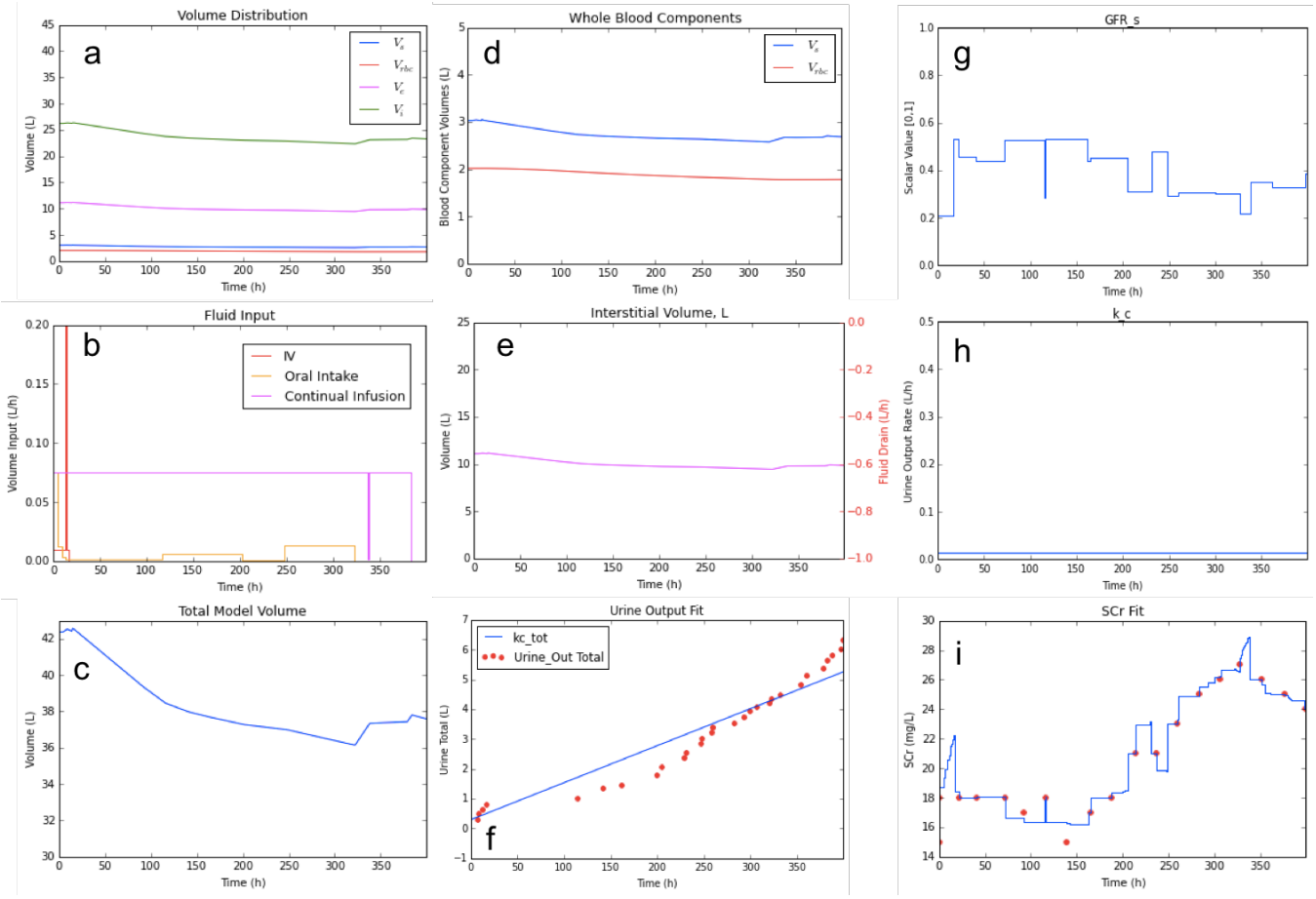

Figure 3.7: Patient 6 (fatality) data and fitted results. Time in hospital = 399 h.

Similar to patient 6, patient 7 (Figure 3.8) was in the hospital for 398 h.This patient received fluid via six methods: IV, oral intake, continual infusion, blood products / colloids, feeding tube, and enteral intake. $k_{c,vol}$ predicts this patient to have fluid levels greater than

100% upon their arrival to the hospital. Patient 6 continually lost fluid via the interstitial compartment which caused a decrease in fluid levels during their time in the hospital. At time points 180 h, 210 h, 260 h, and 300 h, significant fluid was lost through $V_e(t)$. To initially counter these losses, IV was administered at an increased rate. At t = 260 h, significant fluid loss in $V_e(t)$ was countered by blood products / colloids administered at a rate of 1.15 L h^{-1}. The total change in SCr is 27 mg L^{-1}. The data shows a gradual increase in SCr over time. The model does not fit SCr values well for the first 100 h which is problematic. This may be due to $GFR_s(t)$ being zero at t = 0 h and creating a significant accumulation of creatinine in a short amount of time. The amount of oscillation present in $GFR_s(t)$ over time is not ideal, however, its ability to fit UOT over time was good. Likewise, SCr values were fit well at t > 100 h. SCr and UOT RMSE values for patient 7 are 5.35 mg L^{-1} and 2.08 L, respectively. The RMSE for SCr was the highest value recorded for all 10 patients and is attributed to the performance of the model over the first four days. UOT RMSE was also exceptionally high, but relatively similar to patients 8 and 9. Underlying conditions for patient 7 not captured by the model may have influenced the results seen here. Increasing the limitations of $GFR_s(t)$ would also improve the fit performance.

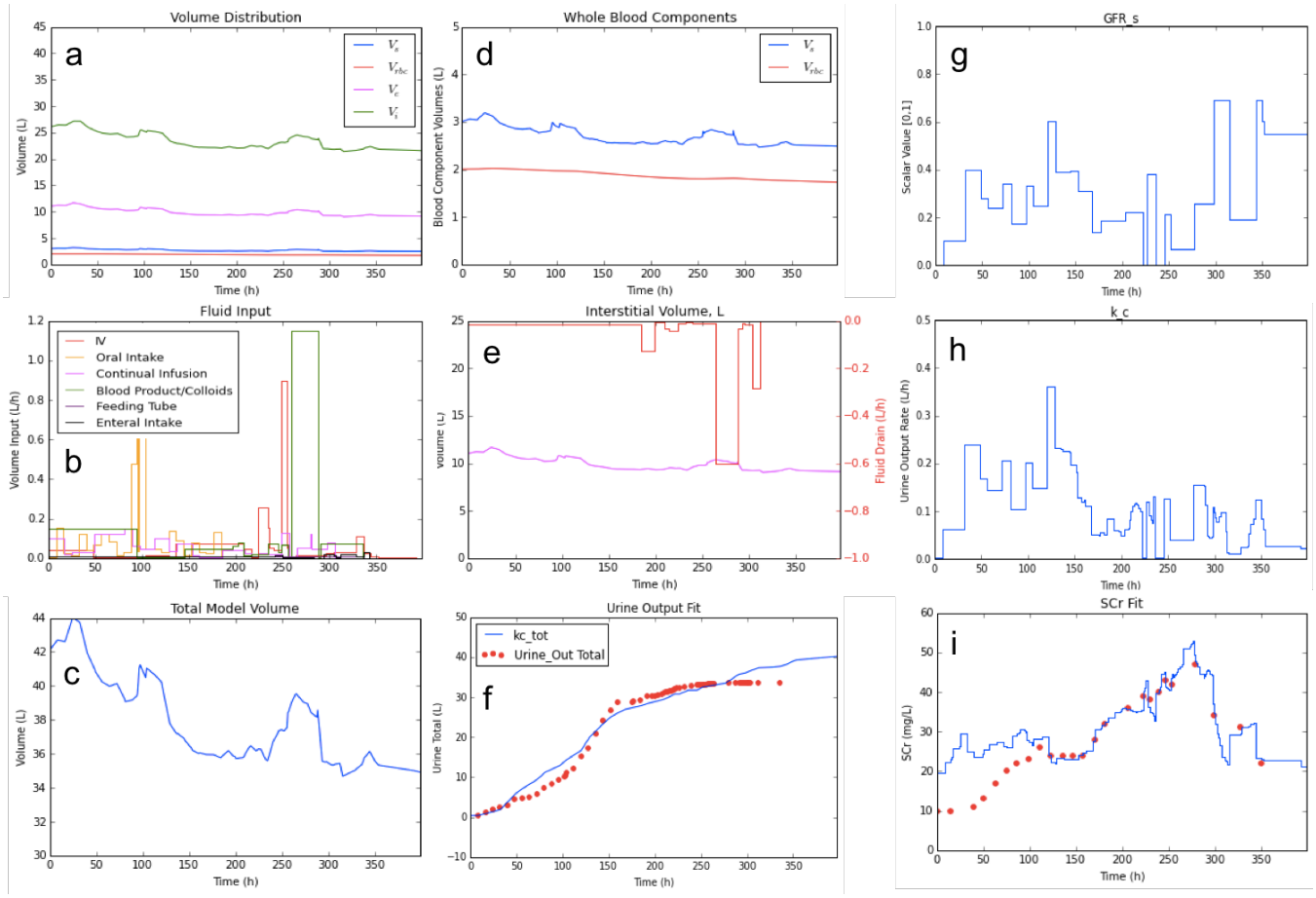

Figure 3.8: Patient 7 (fatality) data and fitted results. Time in hospital = 398 h.

Patient 8 was in the hospital for 470 hours (19.5 days). The results of the fit for this patient are given in Figure 3.9. Seven methods of fluid input were utilized: IV, oral intake, continual infusion, blood products / colloids, operation intake, feeding tube, and enteral intake. Enteral intake was steady over the first 400 hours of their stay and then decreased. A feeding tube was utilized after their first 130 hours in the hospital. interstitial fluid was continually lost during their time. Highly notable losses of fluid occurred at t = 180 h, 350 h, and 400 h. Its impact on the interstitial compartment and $V_{tot}(t)$ are apparent. The urine output rate over the first 10 days in the hospital was very low and the model was not capable of capturing it. An underlying condition in this patient is not accounted for by the

model. The model focused on minimizing the error in SCr as a result of the poor fit to UOT. The total change in SCr was 52 mg L^{-1}. For patient 8, SCr and UOT RMSE values are 1.16 mg L^{-1} and 1.84 L. This fit could be improved upon in the future.

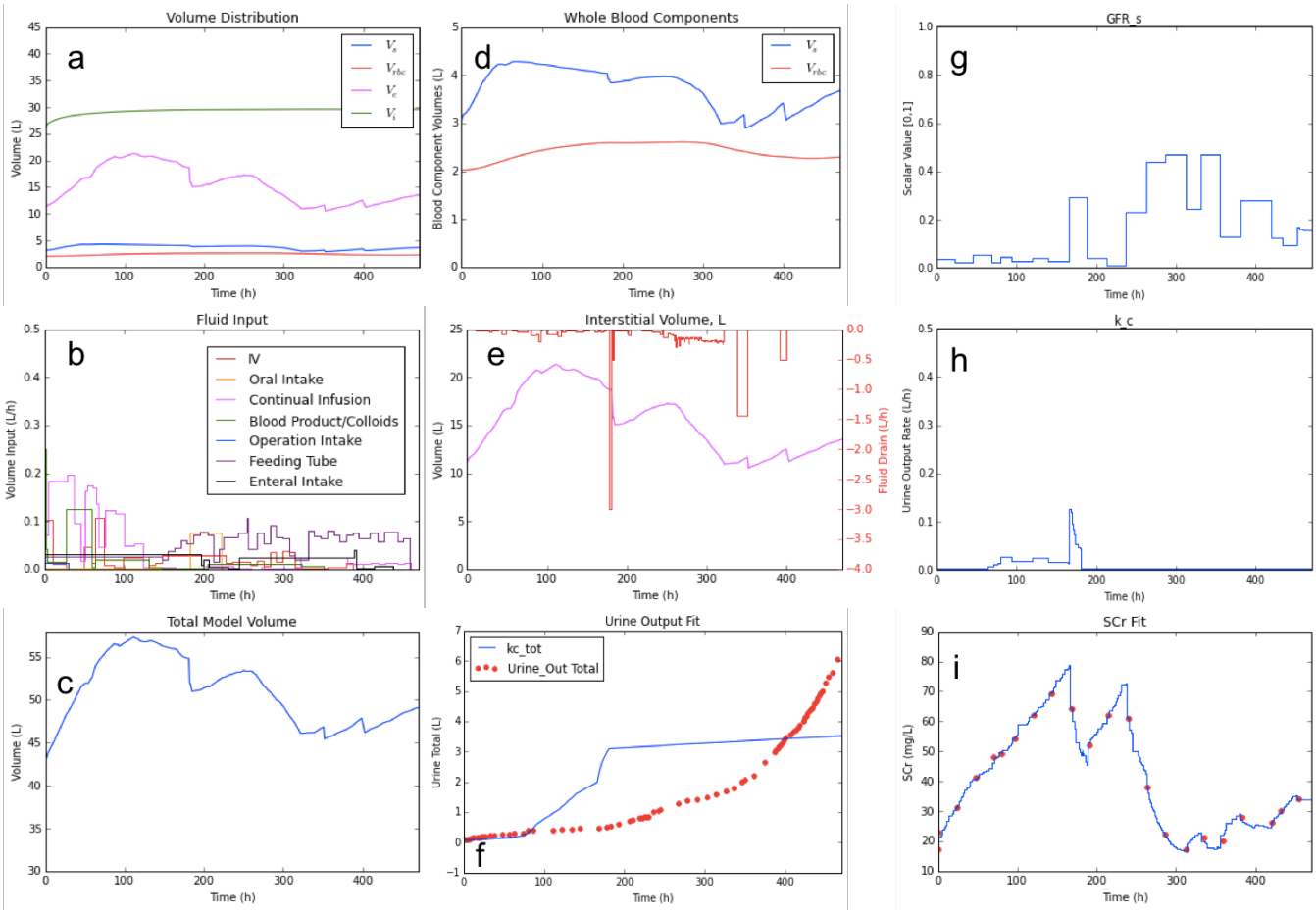

Figure 3.9: Patient 8 (fatality) data and fitted results. Time in hospital = 470 h.

Firgure 3.10 gives the fitted results for patient 9 during their 378 hours in the hospital. Five different methods of fluid delivery were used for this individual: IV, oral intake, continual infusion, feeding tube, and enteral intake. After t = 240 h, the only form of fluid delivery was oral intake. Upon admittance, significant rates of fluid delivery were administered to the patient via IV. The interstitial compartment was losing significant amounts of fluid at the same time. $k_{c,vol}$ indicated close to 100% fluid hydration at t = 0 h. The significant losses of fluid via $V_e(t)$ countered that hydration level. Upon hospital admission, SCr continually increased and $GFR_s(t)$ ranged from 0.0 to 0.1 over the first 3 days. $GFR_s(t)$ slowly improved

over time. SCr ranged from 17 mg L^{-1} to 92 mg L^{-1}. The effectiveness of the model to capture the data for patient 9 was generally good. SCr and UOT RMSE values were 5.67 mg L^{-1} and 1.84, respectively. But, the fairly high RMSE for SCr is due to two data points missed. The rest of the fit to SCr was good. $GFR_s(t)$ varied more than what is desirable, however, the total variation was less then what is seen for other patients.

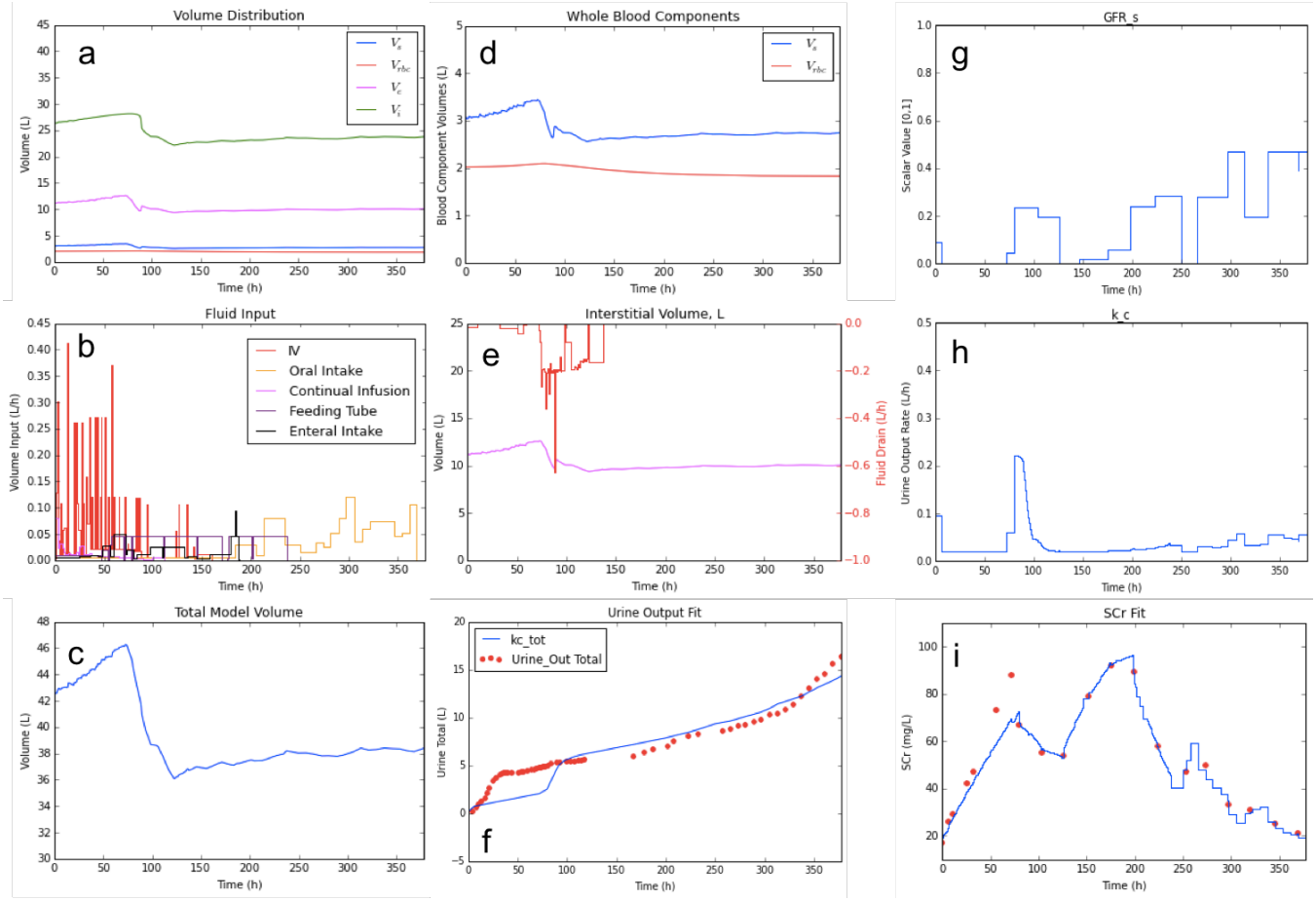

Figure 3.10: Patient 9 (fatality) data and fitted results. Time in hospital = 378 h.

Over the course of 394 hours, patient 10 (Figure 3.11) received three total fluid types: IV, oral intake, and continual infusion. The rate of continual infusion was significant increased over the course of a day. This significant fluid increase corresponded to a significant, continual loss of interstitial fluid from t = 75 h to t = 180 h. The range of SCr values is from 9 mg L^{-1} to 93 mg L^{-1}. The model did not properly fit UOT. From t = 100 h to t > 350 h, the urine production rate was very minimal and the model was not capable of fitting this very small

increase. An underlying condition present in patient 10 not captured by the framework of the model is responsible for this missed ability to capture the results. Due to this, the model assumed severe dehydration and fit the SCr data with $GFR_s(t)$. At the same time point that SCr data began to increase, urine production significantly decreased (nearly nothing at all). This may infer the onset of anuria as the sharp increase in SCr would conclude highly minimal kidney function. A model capable of capturing the development of anuria would be capable of fitting this data set. $GFR_s(t)$ varied with SCr more than what would be desired. Overall, this fit indicates an opportunity to further develop this model.

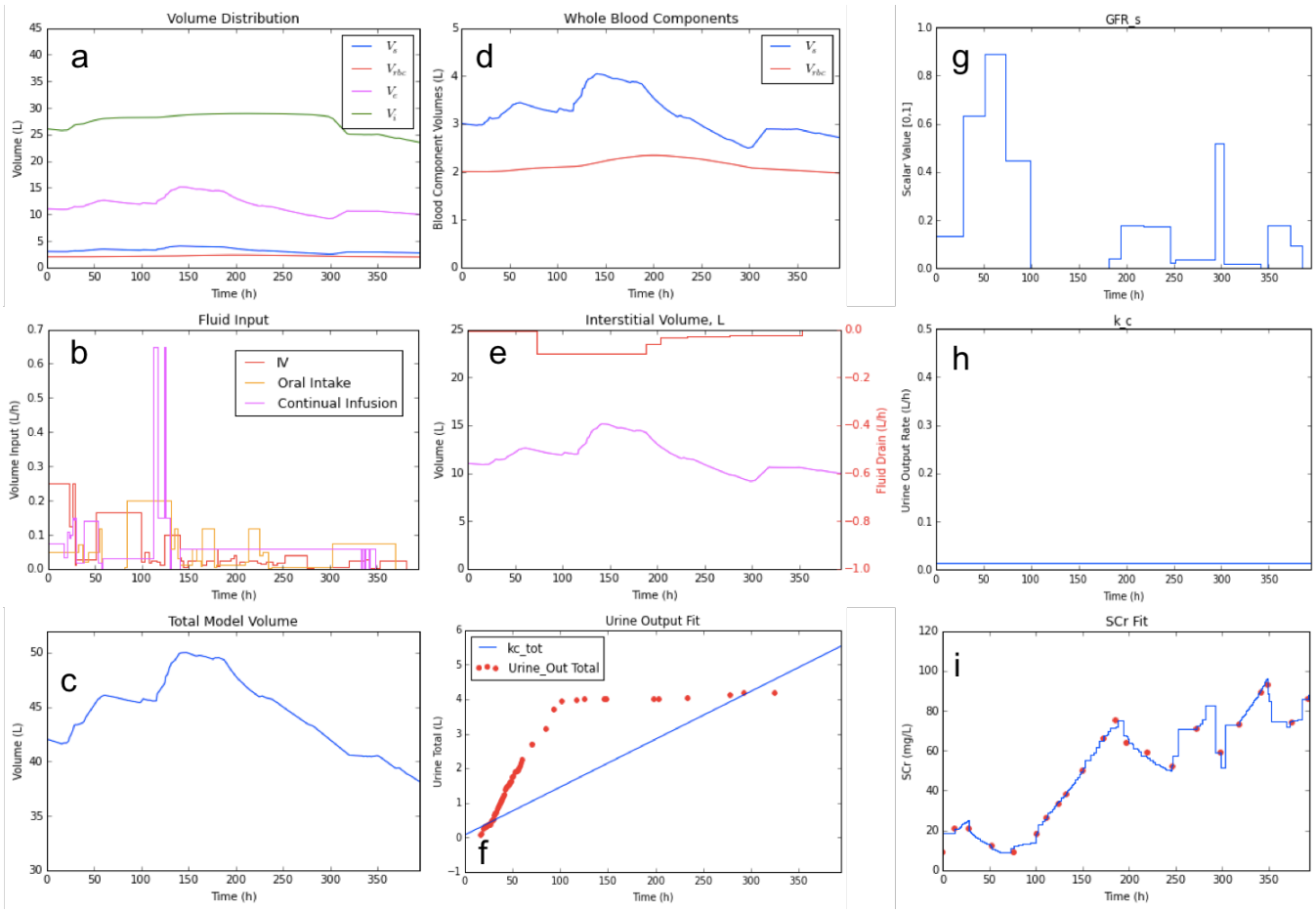

Figure 3.11: Patient 10 (fatality) data and fitted results. Time in hospital = 394.

Table 3.2: Patient data fit calculations SSE, MSE, and RMSE

Patient #	SSE SCr ($mg^2\ L^{-2}$)	MSE SCr ($mg^2\ L^{-2}$)	RMSE SCr ($mg\ L^{-1}$)	SSE UOT (L^2)	MSE UOT (L^2)	RMSE UOT (L)
1	22.75	1.75	1.32	38.05	0.67	0.81
2	0.18	0.01	0.11	3.61	0.36	0.60
3	84.27	3.83	1.96	6.82	0.11	0.33
4	516.11	19.11	4.37	1279.78	9.77	3.12
5	48.74	1.87	1.36	63.62	0.41	0.64
6	3.38	0.18	0.42	8.00	0.28	0.53
7	714.43	28.57	5.35	268.87	4.34	2.08
8	29.48	1.34	1.16	384.70	3.37	1.84
9	664.02	32.20	5.67	200.17	3.39	1.84
10	56.14	2.55	1.60	93.22	1.73	1.31

Table 3.3: Patient k_c parameter fit results

Patient #	$k_{c,vol}$	$k_{c,min}$	$k_{c,max}$	$k_{c,slope}$
1	42.54	0.001	0.600	-1.5
2	55.20	0.015	0.600	-3.0
3	55.19	0.020	0.600	-3.0
4	25.82	0.02	1.200	-1.5
5	44.90	0.020	1.200	-1.5
6	50.86	0.012	0.898	-2.6
7	35.83	0.001	0.600	-3.0
8	55.19	0.001	0.600	-3.0
9	39.61	0.020	0.675	-1.5
10	55.16	0.014	0.652	-3.0

3.2.3 Patient Fitting Conclusions

The model demonstrated the ability to capture many attributes of real patient data. The correlation of kidney function impacting both SCr and fluid dynamics was evident in patients at 100% fluid hydration. As expected, patients below 100% fluid hydration did not have changes in urine production rate adjust with SCr measurements. Surviving patients had better model fitting performance. This may be due to underlying conditions that exist with fatality patients. Discovering some of the more prominent conditions in a population and incorporating those conditions into this model may aid clinicians in diagnosing their patients. Developing a more effective way to penalize $GFR_s(t)$ would also enhance the performance of this model. Another effective way to improve this model would be to adjust the initial volumes rather then adjusting $k_{c,vol}$ to capture the initial state of hydration of a patient. This would allow for the physiologically guided development of this model to accurately capture fluid distribution of a patient.

4.0 SUMMARY AND FUTURE WORK

4.1 SUMMARY

In this work, we suggest the importance of a model grounded in physiological principals to more accurately and quickly detect acute kidney injury and help avoid interstitial edema. In Chapter 1, we discuss previously developed models and how they influenced the understanding of AKI. Chapter 2 develops a model structure that captures the four main volume and creatinine compartments within the human anatomy of a 70 kg individual. Under dynamic conditions, rate constants connecting the compartments were given mathematical relationships that were biologically correspondent. Additionally, a non-static approach to creatinine generation and urine production rates were woven into the model. After discussing the model development, sixteen hypothetical scenarios were drawn to give possible incidences that may occur in the ICU. $GFR_s(t)$ outcomes were predicted with a model that considers only creatinine dynamics versus a model considering volume, creatinine, and creatinine generation dynamics. The outcomes of this study show how the dynamic model is capable of predicting AKI in situations where a static model fails to identify kidney injury. Proceeding these scenarios, the static and dynamic models were projected over a 24 hour period to demonstrate potential $GFR_s(t)$ decreases required to generate an observed ΔSCr. The results of this study highlights that a static model under-predicts the depth of kidney damage and/or the duration of kidney injury for a given ΔSCr.

The intent of chapter 3 is to detail the strengths and limitations of the model designed in chapter 2. Chapter 3 fits the model to data of ten patients in the ICU at UPMC. One key finding of this chapter is the rate of urine production proceeding kidney function increase. As $GFR_s(t)$ increased, if the patient is hydrated, urine production rate significantly increases.

Identifying physical indicators of kidney recovery such as this one could further aid in the recovery process. Also, in patients that were dehydrated throughout their stay in the ICU, urine production rate remained constant as SCr continually changed. This was the case for patients 2, 3, 5, 6, and 9. The level of correlation between changes in SCr and urine production rate should be used as an indicator that a patient may be in need of additional fluids. Overall, surviving patients fit the model better than patients who did not survive. Outlier conditions present in the non-surviving patients may not be captured by the model which could be a factor that impacts the fits.

4.2 FUTURE WORK

There is a multitude of directions this work could take going forward. Tubular secretion was not accounted for here and could be studied to determine if it would have significant impact on the model. Tubular secretion works via active transport and moves creatinine from the blood plasma to the urinary tract [13]. To verify whether or not the rate of secretion is dynamic as $GFR_s(t)$ changes would lead to further validation of SCr as a key biomarker in the prediction of AKI. Another direction this work could consider is the prediction of an accurate total fluid volume at t=0 h for the model optimization. Currently, the model always initiates at 42 L. However, not all patients are at this level of hydration. The current model seeks to account for the state of dehydration by adjusting $k_{c,vol}$, however, this negates the effectiveness of the dynamic rates between the volumetric compartments. Hematocrit is a data point collected that could be utilized to determine the rate of volume dynamics between $V_b(t)$ and $V_s(t)$. This work studies the physiologies of a 70 kg individual. Considering weights more prone to AKI may be beneficial. Likewise, other attributes such as age, gender, and pre-existing conditions could further improve the usefulness of this system. Patients entering the ICU may have issues with more organs than just the kidneys. Tying this model in with mathematical representations of other organs and how they play off of one another could also be insightful. The rate of creatinine generation in a dehydrated individual was not considered in this paper and may have an impact on SCr measurements if dehydration influences it.

Hormonal influences on volume dynamics were considered, but more research could lead to further model development. Considering the influence of common health issues would be another path to consider.

APPENDIX A

RIFLE AND AKIN DEFINITIONS OF AKI

Table A1: RIFLE Classification of AKI [1] which includes designation for urinary output and SCr. End Stage Kidney Disease is Abbreviated ESKD. Urinary output = UO.

	GFR Criteria	Urine Output Criteria
Risk	Increase SCr x 1.5 or GFR decrease $> 25\%$	UO < 0.5 ml/kg/h x6h
Injury	Increased SCr x2 or GFR decrease $> 50\%$	UO < 0.5 ml/kg/h x 12 h
Failure	Increase SCr x3 GFR decrease 75% OR SCr $\geq$ 4mg/dl	UO < 0.3 ml/kg/h x24 h or Anuria x12 h
Loss	Complete loss of function > 4 weeks	Complete loss of function > 4 weeks
ESKD	End stage kidney disease > 3 months	End stage kidney disease > 3 months

Table A2: AKIN classification of AKI [36]. Modified from RIFLE criteria.

	Creatinine Criteria	Urine Output Criteria
Stage 1	serum creatinine of $\geq$ 0.3 mg/dL ($\geq$ 26.4 umol/L) or increase to $\geq$150% - 200% (1.5 - to - 2-fold) from baseline	$\leq$ 0.5 ml/kg/h for $>$ 6 h
Stage 2	Increase in SCr to $>$ 200% - 300% ($>$ 2 to 3-fold) from baseline	<0.5 ml/kg/h for $>$ 2h
Stage 3	Increase in SCr to $>$ 300% ($>$ 3-fold) from baseline or SCr $\geq$4.0 mg/dL $\geq$354umol/L with an acute rise of at least 0.5 mg/dL (44 umol/L)	<0.3 ml/kg/h or anuria>12h

APPENDIX B

LIBRARY DEVELOPMENT FOR MODEL INITIAL CONDITIONS

```
# -*- coding: utf-8 -*-
"""
Created on Sat Oct 21 23:58:14 2017

@author: Evan
"""
import numpy as np
import pylab as P
import pandas as pd

Continual_Inf = 0.0958
Days = 20

"""
My Volumes at the end
"""
Vs_f = []
Vrbc_f = []
Ve_f = []
Vi_f = []
Cs_f = []
Crbc_f = []
Cxe_f = []
Cxm_f = []
CsW_f = []
CrbcW_f = []
CxW_f = []

gfr_v = np.linspace(0,1,1000)
for i in range(len(gfr_v)):
    def rhs(t,y):
```

```
        Vs, Vrbc, Ve, Vi, Cs, Crbc, Cxe, Cxm, CsW, CrbcW, CxW = y
        GFRs = gfr_v[i]
        kie = 5 / (1 + np.exp(3*(Vi - 26)))
        kei = kie*26/11
        Vtot = Vi+Ve+Vs+Vrbc
        Vb = Vs + Vrbc
        kGFR = 44/(3*8)*GFRs
        kc = 0.0094 + (0.833 - 0.0094)/(1+np.exp(-2.25*(Vtot - (43.1892))))*GFRs
        S = .1/3 / (1+np.exp(-6*(Vtot - 38.2)))
        ksr = .006
        krs = 3/2*ksr
        kse = 44/3 + (1/(1+ 1*np.exp(-6*(Vb - 7))))*2.1429*Ve
        kes = 4
        dVs = krs*Vrbc - ksr*Vs + kes*Ve - kse*Vs - kc - S + Continual_Inf
        dVrbc = ksr*Vs - krs*Vrbc
        dVe = -(kei+kes)*Ve + kie * Vi + kse*Vs
        dVi = kei*Ve - kie*Vi
        G = 30 + (52.9557 - 30)*GFRs/(0.6397 + GFRs)
        Vx = Vi + Ve
        Vxe = 0.6*Vx
        Vxm = 0.4*Vx
        dVxm = 0.4*(dVi + dVe)
        dVxe = 0.6*(dVi + dVe)
        R = 3.12*(0.7*Vrbc/((0.7/0.93)*(Vrbc/(Vs-Vrbc))+1))
        Q = 5
        dCxm = G/Vxm + Q*(Cs-Cxm)/Vxm - Cxm/Vxm*dVxm
        dCxe = Q*(Cs-Cxe)/Vxe - Cxe/Vxe*dVxe
        dCs = Q/Vs*(Cxm - Cs) + Q/Vs*(Cxe - Cs) + R/Vs*(Crbc-Cs) - (kGFR)*Cs -
            Cs/Vs*dVs
        dCrbc = R*(Cs-Crbc)/Vrbc - Crbc/Vrbc*dVrbc

        ######## COMPARATIVE MODEL

        R0 = 1.743#454
        kGFR_W = 44/(3*8)*gfr_v[i]
        dCxW = 5/(11+26)*(CsW-CxW)
        dCsW = 5/3*(CxW-CsW) + R0/3*(CrbcW-CsW) + 44/3 -kGFR_W*CsW
        dCrbcW = R0*(CsW-CrbcW)/2
        return P.array([dVs, dVrbc, dVe, dVi, dCs, dCrbc, dCxe, dCxm, dCsW, dCrbcW,
            dCxW])
Cs0   = 8
Crbc0 = Cs0
Cxe0 = Cs0
Cxm0 = 44/5 + Cxe0
y0=np.array([3, 2, 11, 26, Cs0, Crbc0, Cxe0, Cxm0, Cs0, Crbc0, Cxe0])
t0=0
model = Explicit_Problem(rhs, y0, t0)
```

```
    model.name = 'Linear Test ODE'
    sim = CVode(model)
    t, y = sim.simulate(Days*24)

    Vs_l = y[-1,0]
    Vrbc_l = y[-1,1]
    Ve_l = y[-1,2]
    Vi_l = y[-1,3]
    Cs_l = y[-1,4]
    Crbc_l = y[-1,5]
    Cxe_l = y[-1,6]
    Cxm_l = y[-1,7]
    CsW_l = y[-1,8]
    CrbcW_l = y[-1,9]
    CxW_l = y[-1,10]

    Vs_f.append(Vs_l)
    Vrbc_f.append(Vrbc_l)
    Ve_f.append(Ve_l)
    Vi_f.append(Vi_l)
    Cs_f.append(Cs_l)
    Crbc_f.append(Crbc_l)
    Cxe_f.append(Cxe_l)
    Cxm_f.append(Cxm_l)
    CsW_f.append(CsW_l)
    CrbcW_f.append(CrbcW_l)
    CxW_f.append(CxW_l)

df = pd.DataFrame({'GFRs':gfr_v,'Vs':Vs_f,'Vrbc':Vrbc_f,'Ve':Ve_f,'Vi':Vi_f,'Cs':Cs_f,'Crbc':
    Crbc_f,'Cxe':Cxe_f,'Cxm':Cxm_f})
dfW = pd.DataFrame({'GFRsW':gfr_v,'CsW':CsW_f,'CrbcW':CrbcW_f,'CxW':CxW_f})
df = df.set_index('Cs')
dfW = dfW.set_index('CsW')
df.to_csv('CKD_Conditions_Library_edr.csv')
dfW.to_csv('CKD_Static_Conditions_edr.csv')
```

APPENDIX C

THEORETICAL MODEL SIMULATION CODE

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jan 3 21:05:23 2017

@author: Evan
"""

import pandas as pd
import numpy as np
import pylab as P
import matplotlib.pyplot as plt

lib = pd.read_csv('CKD_Conditions_Library_edr.csv')
lib = lib.set_index('Cs')
libW = pd.read_csv('CKD_Static_Conditions_edr.csv')
libW = libW.set_index('CsW')


Cs0 = 8 # Creatinine Baselines: 8, 12, 21, 38, 8 = 100%, 12 = 66.7%, 21 = 38.1%, 38 =
    21.0%
Injury = 0 #0,0.35,0.7,1.0
lib_index = lib.index.get_loc(Cs0, method = 'nearest')
lib_indexW=libW.index.get_loc(Cs0, method = 'nearest')
cmin = 6 # Building windows for SCr Concentration Viewing
cmax = 18 # None = 30 ; Stage2 = 40 ; Stage3 = 60 ; Stage4 = 90

Inf =0.3125
Days = 3
d_tot = 8

def rhs(t,y):
```

```
Vs, Vrbc, Ve, Vi, Cs, Crbc, Cxe, Cxm, CsW, CrbcW, CxW = y
GFR_s = lib.iloc[lib_index ][ 'GFRs']# - Injury*lib.iloc[lib_index ][' GFRs']*(t>80) #-
    .7#*(t>80)#*(t>48)#/(1+np.exp(-.2*(t - 48))) #- .68*(t>48)#*(t<96)
G = 30 + (52.9557 - 30)*(GFR_s)/(0.6397 + GFR_s)#44
kie = 5 / (1 + np.exp(3*(Vi - 26)))
kei = kie*26/11
Vtot = Vi+Ve+Vs+Vrbc
Vb = Vs + Vrbc
kGFR = 44/(3*8)*GFR_s
kc = 0.0094 + GFR_s*(0.833 - 0.0094)/(1+np.exp(-2.25*(Vtot - (43.1892))))
S = .1/3 / (1+np.exp(-6*(Vtot - 38.2)))
ksr = .006
krs = 3/2*ksr
kse = 44/3 + (1/(1+ 1*np.exp(-6*(Vb - 7))))*2.1429*Ve
kes = 4
dVs = krs*Vrbc - ksr*Vs + kes*Ve - kse*Vs - kc - S#+ 0.0958*(t<24) +0.5*(t>72)*(t
    <96) + 0.0958*(t>=96)#.5*(t>48)*(t<60) + .1166*(t>60)# - S #+ .11923# +
    .5/(1+np.exp(-3*(t - 48))) - .5/(1+np.exp(-3*(t - 60))) + .1166/(1+np.exp(-3*(t
    - 60)))# .5*(t>48)#*(t<60) + .1166*(t>60)
dVrbc = ksr*Vs - krs*Vrbc
dVe = -(kei+kes)*Ve + kie * Vi + kse*Vs
dVi = kei*Ve - kie*Vi
Vx = Vi + Ve
Vxe = 0.6*Vx
Vxm = 0.4*Vx
dVxm = 0.4*(dVi + dVe)
dVxe = 0.6*(dVi + dVe)
R = 3.12*(0.7*Vrbc/((0.7/0.93)*(Vrbc/(Vs-Vrbc))+1))
Q = 5

dCxm = G/Vxm + Q*(Cs-Cxm)/Vxm - Cxm/Vxm*dVxm
dCxe = Q*(Cs-Cxe)/Vxe - Cxe/Vxe*dVxe
dCs = Q/Vs*(Cxm - Cs) + Q/Vs*(Cxe - Cs) + R/Vs*(Crbc-Cs) - (kGFR)*Cs - Cs/
    Vs*dVs
dCrbc = R*(Cs-Crbc)/Vrbc - Crbc/Vrbc*dVrbc
#Static Model
GFRsW = libW.iloc[lib_indexW]['GFRsW'] - Injury*libW.iloc[lib_indexW]['GFRsW']*(t
    >80)
R0 = 1.743
kGFR_W = 44/(3*8)*GFRsW
dCxW = 5/(11+26)*(CsW-CxW)
dCsW = 5/3*(CxW-CsW) + R0/3*(CrbcW-CsW) + 44/3 -kGFR_W*CsW
dCrbcW = R0*(CsW-CrbcW)/2

return P.array([dVs, dVrbc, dVe, dVi, dCs, dCrbc, dCxe, dCxm, dCsW, dCrbcW, dCxW])
```

```
Crbc0 = Cs0
Cxe0 = Cs0
Cxm0 = lib.iloc[lib_index]['Cxm']
y0=np.array([lib.iloc[lib_index]['Vs'], lib.iloc[lib_index]['Vrbc'], lib.iloc[lib_index]['Ve'],
    lib.iloc[lib_index]['Vi'], Cs0, Crbc0, Cxe0, Cxm0, Cs0, libW.iloc[lib_indexW]['CrbcW'],
    libW.iloc[lib_indexW]['CxW']])
t0=0

from assimulo.problem import Explicit_Problem
model = Explicit_Problem(rhs, y0, t0)
model.name = 'Linear Test ODE'
from assimulo.solvers import CVode
sim = CVode(model)

tfinal = 28 - d_tot*2 + 2*(d_tot - 1)*8/d_tot + 24*(Days - 1)
t, y = sim.simulate(2)

ymaster=[]
ymaster.append(y[:,:])
tmaster =[]
tmaster.append(t[:])

tspan = []
Drink = []
for i in range(Days):
    for m in range(d_tot):
        if m < (d_tot - 1):
            Drink = [(2+ 2*m*8/d_tot + 24*i, 4 + 2*m*8/d_tot + 24*i )]
        else:
            Drink = [(2 + 2*m*8/d_tot + 24*i, 28 - d_tot*2 + 2*m*8/d_tot + 24*i)]
        tspan.append(Drink)
        tspan_ar = sum(tspan,[])

for tspan in tspan_ar:
    y0 = [y[-1,0]+Inf, y[-1,1], y[-1,2], y[-1,3], y[-1,4]*y[-1,0] / (y[-1,0]+Inf), y[-1,5], y
        [-1,6], y[-1,7], y[-1,8], y[-1,9], y[-1,10]]
    model = Explicit_Problem(rhs, y0, tspan[0])
    sim = CVode(model)
    t,y = sim.simulate(tspan[1])
    ymaster.append(y[:,:])
    tmaster.append(t[:])

y0 = y[-1,:]
model = Explicit_Problem(rhs, y0, tspan[-1])
t,y = sim.simulate(tfinal)
ymaster.append(y)
```

```
tmaster.append(t)

ymaster = np.concatenate(ymaster, axis=0)
tmaster = np.concatenate(tmaster, axis=0)
"""
Graphs
"""
Vs = ymaster[:,0]
Vrbc = ymaster[:,1]
Ve = ymaster[:,2]
Vi = ymaster[:,3]
Cs = ymaster[:,4]
Crbc = ymaster[:,5]
Cxe = ymaster[:,6]
Cxm = ymaster[:,7]
CsW = ymaster[:,8]
CrbcW = ymaster[:,9]
CxW = ymaster[:,10]
Vb = Vs + Vrbc
Vtotal = Vs+Vrbc+Ve+Vi
"""
Volume
"""
tf = Days * 24
P.subplot(4,1,1)
P.plot(tmaster,Vi, label='$V_i$', linewidth = 2, color = 'magenta')
P.title('Volume Kinetics')
P.legend()
P.ylim(25,27)
P.xlim(0,tf)
P.locator_params(axis='y', nbins=2)
P.gca().xaxis.set_major_locator(P.NullLocator())

P.subplot(4,1,2)
P.plot(tmaster,Ve, label='$V_e$', linewidth = 2, color = 'green')
P.ylim(10,13)
P.xlim(0,tf)
P.locator_params(axis='y', nbins=2)
P.gca().xaxis.set_major_locator(P.NullLocator())
P.legend()

P.subplot(4,1,3)
P.plot(tmaster,Vs, label='$V_s$', linewidth = 2, color = 'blue')
P.ylim(2,4)
P.xlim(0,tf)
P.locator_params(axis='y', nbins=2)
P.gca().xaxis.set_major_locator(P.NullLocator())
```

```
P.legend()

P.subplot(4,1,4)
P.plot(tmaster,Vrbc, label='$V_{rbc}$', linewidth = 2, color = 'red')
P.ylim(1,3)
P.xlim(0,tf)
P.xticks(P.arange(0,24*(Days+1),24))
P.locator_params(axis='y', nbins=2)
P.legend()
P.xlabel('time (h)')

P.figure()
P.ylabel('Volume (L)')
P.xlabel('Time (h)')
P.plot(tmaster,Vs, label='$V_s$', linewidth = 2, color = 'blue')
P.plot(tmaster,Vrbc, label='$V_{rbc}$', linewidth = 2, color = 'red')
P.plot(tmaster,Ve, label='$V_e$', linewidth = 2, color = 'green')
P.plot(tmaster,Vi, label='$V_i$', linewidth = 2, color = 'magenta')
P.ylim((0,30))
P.xlim((0,tf))
P.grid()
P.legend(loc = 1, bbox_to_anchor=(1.27, 0.8))
vlines = [24,48,72,96, 120, 144,168,192,216,240]
for xpos in vlines:
    P.axvline(x = xpos, color = 'black')
P.xticks(P.arange(0,24*(Days+1),24))
"""
Concentration
"""
P.figure()
P.ylabel('Concentration (mg/L)')
P.xlabel('Time (h)')
P.plot(tmaster,Cs, label='Cs(t)', linewidth = 2, color = 'blue')
P.plot(tmaster,Crbc, label='Crbc(t)', linewidth = 2, color = 'red')
P.plot(tmaster,Cxm, label='Cxm(t)', linewidth = 2, color = 'green')
P.plot(tmaster,Cxe, label='Cxe(t)', linewidth = 2, color = 'magenta')

#P.plot(tmaster,CsW, label='SCr (Static Volume and Generation)', linewidth = 2, color = '
    black', linestyle = '--')

P.ylim((cmin,cmax))
P.xlim((0,tf))
#P.legend(loc=2)
P.legend(loc = 2, bbox_to_anchor=(0, .75), ncol=1)
ylines = [Cs0, np.interp(Days*24, tmaster, Cs)]
for xpos in vlines:
    P.axvline(x = xpos, color = 'black')
```

```
P.xticks(P.arange(0,24*(Days+1),24))
P.grid()

"""
Total Volume
"""
P.figure()
P.ylabel('Total Volume (L)')
P.xlabel('Time (h)')
P.plot(tmaster,Vtotal, label='Vtotal', linewidth = 2, color = 'red')
P.ylim((35,45))
P.xlim((0, tf))
ytotlines = [38.22,40.74,42]
for xpos in ytotlines:
    P.axhline(y = xpos, color = 'black')
for xpos in vlines:
    P.axvline(x = xpos, color = 'black')
P.xticks(P.arange(0,24*(Days+1),24))
"""
Blood/ Interstitial Volume
"""
P.figure()
P.ylabel('Vb (L)')
P.xlabel('Ve (L)')
P.plot(Ve,Vb, linewidth = 2, color = 'blue')
P.xlim(11,42)
P.ylim(5,8)
P.axvline(x = 11)
blines = [5,7]
for yspot in blines:
    P.axhline(y = yspot)

"""
Extracting Information from Plots
"""

O = '\033[33m' # orange
P = '\033[35m' # purple
W = '\033[0m'  # white (normal)
print('Vs @ 120h = %s'%(np.interp(120, tmaster, Vs)))
print('Vrbc @ 120h = %s'%(np.interp(120, tmaster, Vrbc)))
print('Ve @ 120h = %s'%(np.interp(120, tmaster, Ve)))
print('Vi @ 120h = %s'%(np.interp(120, tmaster, Vi)))
print(P+'Vtot @ 0h = %s'%(np.interp(0,tmaster,Vtotal)))
print(P+'Vtot @ 104h = %s'%(np.interp(104,tmaster,Vtotal)))
print(P+'SCr @ 120h = %s'%(np.interp(120, tmaster, Cs)))
print(O+'SCrW @ 120h = %s'%(np.interp(120, tmaster, CsW)))
```

```
print(W+'SCr:SCrW Ratio @ 120h = %s'%(np.interp(120, tmaster, Cs)/np.interp(120, tmaster,
    CsW)))
print(P+'GFRs @t=0h = %s' %(lib.iloc[lib_index]['GFRs']))
print(P+'GFRs @t>80h = %s'%(lib.iloc[lib_index]['GFRs'] - Injury*lib.iloc[lib_index]['GFRs']))
print(O+'GFRsW @t=0h = %s' %(libW.iloc[lib_indexW]['GFRsW']))
print(O+'GFRsW @t>80h = %s'%(libW.iloc[lib_indexW]['GFRsW'] - Injury*libW.iloc[
    lib_indexW]['GFRsW']))
print(P+'Change in SCr from 80h to 104h = %s'%(np.interp(104, tmaster, Cs)-np.interp(80,
    tmaster, Cs)))
print(O+'Change in SCrW from 80h to 104h = %s'%(np.interp(104, tmaster, CsW)-np.interp
    (80, tmaster, CsW)))
if Injury == 0:
    print(W+'%% difference in changes = N/A')
else:
    print(W+'%% difference in changes = %s'%((-(np.interp(104, tmaster, Cs)-np.interp(80,
        tmaster, Cs)) + (np.interp(104, tmaster, CsW)-np.interp(80, tmaster, CsW)))/(np.
        interp(104, tmaster, CsW)-np.interp(80, tmaster, CsW))*100))
print(' difference  in changes = %s'%((np.interp(104, tmaster, Cs)-np.interp(80, tmaster, Cs))
    - (np.interp(104, tmaster, CsW)-np.interp(80, tmaster, CsW))))
print(P+'Change in SCr from 72h to 96h = %s'%(np.interp(96, tmaster, Cs)-np.interp(72,
    tmaster, Cs)))
print(O+'Change in SCrW from 72h to 96h = %s'%(np.interp(96, tmaster, CsW)-np.interp(72,
     tmaster, CsW)))
if Injury == 0:
    print(W+'%% difference in changes = N/A')
else:
    print(W+'%% difference in changes = %s'%(((np.interp(96, tmaster, CsW)-np.interp(72,
        tmaster, CsW))-(np.interp(96, tmaster, Cs)-np.interp(72, tmaster, Cs)))/(np.interp(96,
         tmaster, CsW)-np.interp(72, tmaster, CsW))*100))
print(' difference  in changes = %s'%((np.interp(96, tmaster, Cs)-np.interp(72, tmaster, Cs)) -
    (np.interp(96, tmaster, CsW)-np.interp(72, tmaster, CsW))))
```

APPENDIX D

MODEL FITTING TO PATIENT DATA

```
# -*- coding: utf-8 -*-
"""
Created on Sat Aug 5 23:12:05 2017

@author: Evan
"""
import os
clear = lambda: os.system('cls')
import matplotlib.pyplot as plt
from pyomo.environ import *
import numpy as np
from pyomo.dae import *
import pandas as pd

PVIDs =
    [34648440,34507970,33981050,34396040,34276822,32914196,33182922,33341773,33164780,32849855]


# Load the CSV's
df_all_1 = pd.read_csv('Patients_Data_SCr.csv')
df_all_2 = pd.read_csv('Patients_Data_FluidOUT.csv')
df_all_3 = pd.read_csv('Patients_Data_FluidIN.csv')
lib = pd.read_csv('CKD_Conditions_Library_edr.csv')
lib = lib.set_index('Cs')
#Processing Per Patient
for NUMBER in PVIDs:
    """
    PROCESSING DATA
    """
    df1 = df_all_1.groupby('PVID').get_group(NUMBER)
    df2 = df_all_2.groupby('PVID').get_group(NUMBER)
```

```
df3 = df_all_3.groupby('PVID').get_group(NUMBER)

df1['Date'] = pd.to_datetime(df1['Date'])                    # Convert string to datetime
    object
starttime = df1.iloc[0]['Date']                              # Get patient start time
elapsed = df1['Date'] - starttime                            # Get time elapsed (pandas timedelta)
elapsed = elapsed/ np.timedelta64(1, 'h')                    # Convert pandas timedelta into
    hours (float)
df1['t'] = elapsed
df1['SCr'] = df1['SCr mg/L']
df1 = df1.set_index('t')
df_SCr = df1
df_SCr.dropna(axis = 0, inplace= False)
df_SCr['diff_SCr'] = df_SCr['SCr'].diff()
df_SCr['diff_SCr'].replace(to_replace = ['NaN'], value = 0, inplace = True)
df_SCr['diff_SCr'] = df_SCr['diff_SCr'].shift(-1)
df_SCr['diff_SCr'].replace(to_replace = ['NaN'], value = 0, inplace = True)
df_SCr['diff_t'] = df_SCr.index.to_series().diff()
df_SCr['diff_t'] = df_SCr['diff_t'].shift(-1)
df_SCr['diff_t'].replace(to_replace = ['NaN'], value = 1, inplace = True)
df_SCr['dSCr'] = df_SCr['diff_SCr']/df_SCr['diff_t']
df_SCr['MAx_24'] = df_SCr['dSCr']*24

df2['Date'] = pd.to_datetime(df2['Date'])                    # Convert string to datetime object
elapsed2 = df2['Date'] - starttime                           #df2.iloc[0]['Datetime']        #
    Calculate time elapsed
elapsed2 = elapsed2/np.timedelta64(1, 'h')                   # Convert time elapsed into
    hours
df2['t'] = elapsed2
df2 = df2.set_index('t')                                     # Set time as index

df3['Date'] = pd.to_datetime(df3['Date'])                    # Convert string to datetime object
elapsed3 = df3['Date'] - starttime                           #df2.iloc[0]['Datetime']        #
    Calculate time elapsed
elapsed3 = elapsed3/np.timedelta64(1, 'h')                   # Convert time elapsed into
    hours
df3['t'] = elapsed3
df3 = df3.set_index('t')


df1 = df1.loc[np.union1d(df1.index.values, df2.index.values)] #df1 is used to give all time
    points to the model
df1 = df1.loc[np.union1d(df1.index.values, df3.index.values)] #so all time points from df1,
    df2, and df3 are given to df1
df1['PVID'].replace(to_replace = ['NaN'], value = NUMBER, inplace = True)
df1['SCr'].interpolate(method='index', axis=0, limit=None, inplace=True, limit_direction='
    forward')
```

```
df1['SCr mg/L'].interpolate(method='index', axis=0, limit=None, inplace=True,
    limit_direction='forward')
df1 = df1.groupby(df1.index).mean()

Urineout = df2.groupby('IODetail').get_group('Urineout')
Urineout = Urineout[Urineout['Volume L'] != 0]
Urineout = Urineout.groupby(Urineout.index).agg(sum)
Urineout['Urine Tot'] = Urineout['Volume L'].cumsum()

df2 = df2.loc[np.union1d(df2.index.values, df1.index.values)]    # df2 time points may not
    exist in df1. Rectify that.
df2 = df2.loc[np.union1d(df2.index.values, df3.index.values)]
df2['PVID'].replace(to_replace = ['NaN'], value = NUMBER, inplace = True)
df2['Volume L'].replace(to_replace = ['NaN'], value = 0, inplace = True)
df2['IODetail']. fillna (value = 'VeLoss', inplace = True)



VeLoss = df_all_2.groupby('PVID').get_group(NUMBER)
VeLoss['Date'] = pd.to_datetime(VeLoss['Date'])          # Convert string to datetime object
elapsed_V = VeLoss['Date'] - starttime                   #df2.iloc [0]['  Datetime']        #
    Calculate time elapsed
elapsed_V = elapsed_V/np.timedelta64(1, 'h')             # Convert time elapsed
    into hours
VeLoss['t'] = elapsed_V
VeLoss = VeLoss.set_index('t')
VeLoss['PVID'].replace(to_replace = ['NaN'], value = NUMBER, inplace = True)
VeLoss['Volume L'].replace(to_replace = ['NaN'], value = 0, inplace = True)
VeLoss['IODetail']. fillna (value = 'VeLoss', inplace = True)
if len(Urineout['Volume L']) == len(VeLoss['Volume L']):
    VeLoss['Volume L'] = 0.0
else:
    VeLoss  = VeLoss.groupby('IODetail').get_group('VeLoss')
VeLoss  = VeLoss.groupby(VeLoss.index).agg(sum)
VeLoss['vol tot'] = VeLoss['Volume L'].cumsum()
VeLoss['diff'] = VeLoss['vol tot']. diff ()
VeLoss['diff']. replace(to_replace = ['NaN'], value = 0, inplace = True)
VeLoss['diff_t'] = VeLoss.index.to_series(). diff ()
VeLoss['diff_t']. replace(to_replace = ['NaN'], value = 1, inplace = True)
VeLoss['slope'] = VeLoss['diff']/VeLoss['diff_t']
VeLoss['slope'] = VeLoss['slope']. shift (-1)
VeLoss['slope']. replace(to_replace = ['NaN'], value = 0, inplace = True)

"""
BEGIN FLUID IN WORK
"""
```

```
Data = []
Data.insert(0, {'PVID' : NUMBER , 'Volume L' : 0})
check_1 = 'IV' in df3.Fluid_In_Method.values
if check_1 == 1:
    IV = df3.groupby('Fluid_In_Method').get_group('IV')
    IV = IV.groupby(IV.index).agg(sum)
    IV.dropna(axis = 0, inplace= False)
    if IV.index[0] > 0:
        IV = pd.concat([pd.DataFrame(Data), IV], ignore_index=False)
    IV['vol tot'] = IV['Volume L'].cumsum()
    IV[' diff '] = IV['vol tot'].diff()
    IV[' diff '].replace(to_replace = ['NaN'], value = 0, inplace = True)
    IV[' diff_t '] = IV.index.to_series().diff()
    IV[' diff_t '].replace(to_replace = ['NaN'], value = 1, inplace = True)
    IV['slope'] = IV[' diff ']/IV[' diff_t ']
    IV['slope'] = IV['slope'].shift(-1)
    IV['slope'].replace(to_replace = ['NaN'], value = 0, inplace = True)

check_2 = 'Oral Intake' in df3.Fluid_In_Method.values
if check_2 == 1:
    Oral_In = df3.groupby('Fluid_In_Method').get_group('Oral Intake')
    Oral_In = Oral_In.groupby(Oral_In.index).agg(sum)
    Oral_In.dropna(axis = 0, inplace= False)
    if Oral_In.index[0] > 0:
        Oral_In = pd.concat([pd.DataFrame(Data), Oral_In], ignore_index=False)
    Oral_In['vol tot'] = Oral_In['Volume L'].cumsum()
    Oral_In[' diff '] = Oral_In['vol tot'].diff()
    Oral_In[' diff '].replace(to_replace = ['NaN'], value = 0, inplace = True)
    Oral_In[' diff_t '] = Oral_In.index.to_series().diff()
    Oral_In[' diff_t '].replace(to_replace = ['NaN'], value = 1, inplace = True)
    Oral_In['slope'] = Oral_In[' diff ']/Oral_In[' diff_t ']
    Oral_In['slope'] = Oral_In['slope'].shift(-1)
    Oral_In['slope'].replace(to_replace = ['NaN'], value = 0, inplace = True)

check_3 = 'Continuous Infusions' in df3.Fluid_In_Method.values
if check_3 == 1:
    Cont_Inf = df3.groupby('Fluid_In_Method').get_group('Continuous Infusions')
    Cont_Inf = Cont_Inf.groupby(Cont_Inf.index).agg(sum)
    Cont_Inf.dropna(axis = 0, inplace= False)
    if Cont_Inf.index[0] > 0:
        Cont_Inf = pd.concat([pd.DataFrame(Data), Cont_Inf], ignore_index=False)
    Cont_Inf['vol tot'] = Cont_Inf['Volume L'].cumsum()
    Cont_Inf[' diff '] = Cont_Inf['vol tot'].diff()
    Cont_Inf[' diff '].replace(to_replace = ['NaN'], value = 0, inplace = True)
    Cont_Inf[' diff_t '] = Cont_Inf.index.to_series().diff()
    Cont_Inf[' diff_t '].replace(to_replace = ['NaN'], value = 1, inplace = True)
    Cont_Inf['slope'] = Cont_Inf[' diff ']/Cont_Inf[' diff_t ']
```

```
    Cont_Inf['slope'] = Cont_Inf['slope'].shift(-1)
    Cont_Inf['slope'].replace(to_replace = ['NaN'], value = 0, inplace = True)

check_4 = 'Blood Products/Colloids' in df3.Fluid_In_Method.values
if check_4 == 1:
    B_prod_col = df3.groupby('Fluid_In_Method').get_group('Blood Products/Colloids')
    B_prod_col = B_prod_col.groupby(B_prod_col.index).agg(sum)
    B_prod_col.dropna(axis = 0, inplace= False)
    if B_prod_col.index[0] > 0:
        B_prod_col = pd.concat([pd.DataFrame(Data), B_prod_col], ignore_index=False)
    B_prod_col['vol tot'] = B_prod_col['Volume L'].cumsum()
    B_prod_col['diff'] = B_prod_col['vol tot'].diff()
    B_prod_col['diff'].replace(to_replace = ['NaN'], value = 0, inplace = True)
    B_prod_col['diff_t'] = B_prod_col.index.to_series().diff()
    B_prod_col['diff_t'].replace(to_replace = ['NaN'], value = 1, inplace = True)
    B_prod_col['slope'] = B_prod_col['diff']/B_prod_col['diff_t']
    B_prod_col['slope'] = B_prod_col['slope'].shift(-1)
    B_prod_col['slope'].replace(to_replace = ['NaN'], value = 0, inplace = True)

check_5 = 'Operating Room Intake' in df3.Fluid_In_Method.values
if check_5 == 1:
    Oper_In = df3.groupby('Fluid_In_Method').get_group('Operating Room Intake')
    Oper_In = Oper_In.groupby(Oper_In.index).agg(sum)
    Oper_In.dropna(axis = 0, inplace= False)
    if Oper_In.index[0] > 0:
        Oper_In = pd.concat([pd.DataFrame(Data), Oper_In], ignore_index=False)
    Oper_In['vol tot'] = Oper_In['Volume L'].cumsum()
    Oper_In['diff'] = Oper_In['vol tot'].diff()
    Oper_In['diff'].replace(to_replace = ['NaN'], value = 0, inplace = True)
    Oper_In['diff_t'] = Oper_In.index.to_series().diff()
    Oper_In['diff_t'].replace(to_replace = ['NaN'], value = 1, inplace = True)
    Oper_In['slope'] = Oper_In['diff']/Oper_In['diff_t']
    Oper_In['slope'] = Oper_In['slope'].shift(-1)
    Oper_In['slope'].replace(to_replace = ['NaN'], value = 0, inplace = True)

check_6 = 'Feeding Tube Intake' in df3.Fluid_In_Method.values
if check_6 == 1:
    Feed_In = df3.groupby('Fluid_In_Method').get_group('Feeding Tube Intake')
    Feed_In = Feed_In.groupby(Feed_In.index).agg(sum)
    Feed_In.dropna(axis = 0, inplace= False)
    if Feed_In.index[0] > 0:
        Feed_In = pd.concat([pd.DataFrame(Data), Feed_In], ignore_index=False)
    Feed_In['vol tot'] = Feed_In['Volume L'].cumsum()
    Feed_In['diff'] = Feed_In['vol tot'].diff()
    Feed_In['diff'].replace(to_replace = ['NaN'], value = 0, inplace = True)
    Feed_In['diff_t'] = Feed_In.index.to_series().diff()
    Feed_In['slope'] = Feed_In['diff']/Feed_In['diff_t']
```

```
    Feed_In['slope'] = Feed_In['slope'].shift(-1)
    Feed_In['slope'].replace(to_replace = ['NaN'], value = 0, inplace = True)

check_7 = 'Enteral Intake' in df3.Fluid_In_Method.values
if check_7 == 1:
    Enteral_In = df3.groupby('Fluid_In_Method').get_group('Enteral Intake')
    Enteral_In = Enteral_In.groupby(Enteral_In.index).agg(sum)
    Enteral_In.dropna(axis = 0, inplace= False)
    if Enteral_In.index[0] > 0:
        Enteral_In = pd.concat([pd.DataFrame(Data), Enteral_In], ignore_index=False)
    Enteral_In['vol tot'] = Enteral_In['Volume L'].cumsum()
    Enteral_In['diff'] = Enteral_In['vol tot'].diff()
    Enteral_In['diff'].replace(to_replace = ['NaN'], value = 0, inplace = True)
    Enteral_In['diff_t'] = Enteral_In.index.to_series().diff()
    Enteral_In['diff_t'].replace(to_replace = ['NaN'], value = 1, inplace = True)
    Enteral_In['slope'] = Enteral_In['diff']/Enteral_In['diff_t']
    Enteral_In['slope'] = Enteral_In['slope'].shift(-1)
    Enteral_In['slope'].replace(to_replace = ['NaN'], value = 0, inplace = True)

"""
END FLUID IN WORK
"""

# Something used for the Constraints, Fluid Input, VeLoss, and Obj Function
def return_pairs(times):
    return [(times[i], times[i+1]) for (i,time) in enumerate(times[:-1])]
# Initialize Model
model = ConcreteModel()

# Define timepoints
model.t = ContinuousSet(initialize = df1.index.values)
"""
FLUID IN SYSTEM FOR PYOMO
"""
if check_1 == 1:
    def IV_in(t):
        for i, t_tuple in enumerate(return_pairs(IV.index.values)):
            t_lb, t_ub = t_tuple
            if t >= t_lb and t <= t_ub:
                return IV.slope.iloc[i]
        return 0
else:
    def IV_in(t):
        return 0

if check_2 == 1:
    def Oral_in(t):
```

```
        for i, t_tuple in enumerate(return_pairs(Oral_In.index.values)):
            t_lb, t_ub = t_tuple
            if t >= t_lb and t <= t_ub:
                return Oral_In.slope.iloc[i]
        return 0
else:
    def Oral_in(t):
        return 0

if check_3 == 1:
    def Cont_in(t):
        for i, t_tuple in enumerate(return_pairs(Cont_Inf.index.values)):
            t_lb, t_ub = t_tuple
            if t >= t_lb and t <= t_ub:
                return Cont_Inf.slope.iloc[i]
        return 0
else:
    def Cont_in(t):
        return 0

if check_4 == 1:
    def Blood_in(t):
        for i, t_tuple in enumerate(return_pairs(B_prod_col.index.values)):
            t_lb, t_ub = t_tuple
            if t >= t_lb and t <= t_ub:
                return B_prod_col.slope.iloc[i]
        return 0
else:
    def Blood_in(t):
        return 0
if check_5 == 1:
    def Oper_in(t):
        for i, t_tuple in enumerate(return_pairs(Oper_In.index.values)):
            t_lb, t_ub = t_tuple
            if t >= t_lb and t <= t_ub:
                return Oper_In.slope.iloc[i]
        return 0
else:
    def Oper_in(t):
        return 0

if check_6 == 1:
    def Feed_in(t):
        for i, t_tuple in enumerate(return_pairs(Feed_In.index.values)):
            t_lb, t_ub = t_tuple
            if t >= t_lb and t <= t_ub:
                return Feed_In.slope.iloc[i]
```

```
        return 0
else:
    def Feed_in(t):
        return 0
if check_7 == 1:
    def Enteral_in(t):
        for i, t_tuple in enumerate(return_pairs(Enteral_In.index.values)):
            t_lb, t_ub = t_tuple
            if t >= t_lb and t <= t_ub:
                return Enteral_In.slope.iloc[i]
        return 0
else:
    def Enteral_in(t):
        return 0

def Ve_out(t):
    for j, t_tuple in enumerate(return_pairs(VeLoss.index.values)):
        t_lb, t_ub = t_tuple
        if t>= t_lb and t<= t_ub:
            return VeLoss.slope.iloc[j]
    return 0
"""
DEFINING STATE VARIABLES
"""
model.Vs = Var(model.t, within=NonNegativeReals, initialize = 3)
model.Vrbc = Var(model.t, within=NonNegativeReals, initialize = 2)
model.Ve = Var(model.t, within=NonNegativeReals, initialize = 11)
model.Vi = Var(model.t, within=NonNegativeReals, initialize = 26)
model.B_tot = Var(model.t, within =NonNegativeReals, initialize = 0)
model.U_tot = Var(model.t, within =NonNegativeReals, initialize = 0)
model.Cs = Var(model.t, within=NonNegativeReals, initialize = 8)
model.Crbc = Var(model.t, within=NonNegativeReals, initialize = 8)
model.Cxe = Var(model.t, within=NonNegativeReals, initialize = 8)
model.Cxm = Var(model.t, within=NonNegativeReals, initialize = 44/10+model.Cs[0])
model.FI_tot = Var(model.t, within=NonNegativeReals, initialize = 0)
model.FO_tot = Var(model.t, within=NonNegativeReals, initialize = 0)
"""
Define Parameter GFR_s
"""
model.GFR_s = Var(model.t, initialize = 1, bounds = (0,8/(df1.loc[df1.index[0], 'SCr'])))
"""
Defining derivative variables
"""
model.dVs = DerivativeVar(model.Vs, wrt=model.t)
model.dVrbc = DerivativeVar(model.Vrbc, wrt=model.t)
model.dVe = DerivativeVar(model.Ve, wrt=model.t)
model.dVi = DerivativeVar(model.Vi, wrt=model.t)
```

```
model.dCs = DerivativeVar(model.Cs, wrt=model.t)
model.dCrbc = DerivativeVar(model.Crbc, wrt=model.t)
model.dCxe = DerivativeVar(model.Cxe, wrt=model.t)
model.dCxm = DerivativeVar(model.Cxm, wrt=model.t)
model.dB_tot = DerivativeVar(model.B_tot, wrt = model.t)
model.dFI_tot = DerivativeVar(model.FI_tot, wrt = model.t)
model.dFO_tot = DerivativeVar(model.FO_tot, wrt = model.t)

"""
DEFINE INITIAL CONDITIONS
"""
Cs0 = df1.loc[df1.index[0], 'SCr']
lib_index = lib.index.get_loc(Cs0, method = 'nearest')
def _init_conditions(model):
    yield model.Vs[0] == lib.iloc[lib_index]['Vs'] # We make the Fluid In and the Fluid
        Out in the data at t = 0 to be already accounted for in the model
    yield model.Vrbc[0] == lib.iloc[lib_index]['Vrbc']
    yield model.Ve[0] == lib.iloc[lib_index]['Ve']
    yield model.Vi[0] == lib.iloc[lib_index]['Vi']
    yield model.Cs[0] == Cs0
    yield model.Crbc[0] == Cs0
    yield model.Cxe[0] == Cs0
    yield model.Cxm[0] == lib.iloc[lib_index]['Cxm']
    yield model.B_tot[0] == Urineout.loc[Urineout.index[0],'Urine Tot']
    yield model.FO_tot[0] == VeLoss.loc[VeLoss.index[0], 'vol tot']
model.init_conditions = ConstraintList(rule=_init_conditions)
"""
Begin Constraints
"""
def _creat_constraintA(model,t):
    return model.Crbc[t] - model.Cs[t] <= 15
model.creat_constraintA = Constraint(model.t, rule = _creat_constraintA)
def _creat_constraintB(model,t):
    return -15 <= model.Crbc[t] - model.Cs[t]
model.creat_constraintB = Constraint(model.t, rule = _creat_constraintB)
#Limiting GFR_s
def GFRs_Group_lb(t):
    for k, t_tuple in enumerate(return_pairs(df_SCr.index.values)):
        t_lb, t_ub = t_tuple
        if t >= t_lb and t <= t_ub:
            return t_lb
    return 0

def GFRs_Group_ub(t):
    for k, t_tuple in enumerate(return_pairs(df_SCr.index.values)):
        t_lb, t_ub = t_tuple
        if t >= t_lb and t <= t_ub:
```

```
            return t_ub
    return 0

def _GFRs_Constraint_1(model,t):
    if t >= GFRs_Group_lb(t) and t < GFRs_Group_ub(t):
        return model.GFR_s[t] == model.GFR_s[GFRs_Group_lb(t)]
    else:
        return Constraint.Skip
model.GFRs_Constraint_1 = Constraint(model.t, rule = _GFRs_Constraint_1)

def _GFRs_Constraint_2(model,t):
    return abs(model.GFR_s[GFRs_Group_lb(t)] - model.GFR_s[GFRs_Group_ub(t)]) <=
        0.5

model.GFRs_Constraint_2 = Constraint(model.t, rule = _GFRs_Constraint_2)
"""
REGULATING kc
"""
model.kc_max = Var(initialize=0.833, bounds=(0.6,1.2))
model.kc_slope = Var( initialize =-2.25, bounds=(-3,-1.5))
model.kc_vol = Var( initialize  = 38, bounds = (25.1892, 55.1892))
model.kc_min = Var(initialize  = 0.0094, bounds = (0.001,0.02))

model.kc = Var(model.t, within=NonNegativeReals)
def _kc_constrA(model,t):
    Vtot = model.Vi[t]+model.Ve[t]+model.Vs[t]+model.Vrbc[t]
    return model.kc[t] == model.kc_min + model.GFR_s[t]*(model.kc_max)/(1+exp(model.
        kc_slope*(Vtot - (model.kc_vol))))
model.kc_constrA = Constraint(model.t, rule=_kc_constrA)

model.VeLoss = Var(model.t, within=NonNegativeReals)
def _set_FluidOut_data(model,t):
    if t in VeLoss.index.values:
        return model.VeLoss[t] == VeLoss.loc[t, 'Volume L']
    else:
        return model.VeLoss[t] == 0
model.set_FluidOut_data = Constraint(model.t, rule=_set_FluidOut_data)

"""
Bladder, Fluid In, & Ve Out
"""
def _B_tot_eq(model,t):
    return model.dB_tot[t] == model.kc[t]
model.B_tot_eq = Constraint(model.t, rule=_B_tot_eq)

def _dFI_tot_Constr(model,t):
    return model.dFI_tot[t] >= 0
```

```
model.dFI_tot_Constr = Constraint(model.t, rule=_dFI_tot_Constr)

def _dFO_tot_ConstrA(model,t):
    return model.dFO_tot[t] >= 0
model.dFO_tot_ConstrA = Constraint(model.t, rule=_dFO_tot_ConstrA)

def _dFO_tot_ConstrB(model,t):
    if t > VeLoss.index[-1]:
        return model.dFO_tot[t] == 0
    else:
        return model.dFO_tot[t] >= 0
model.dFO_tot_ConstrB = Constraint(model.t, rule=_dFO_tot_ConstrB)
"""
Volume Compartments
"""
def _Vs_eq(model,t): # Fluid is infused into this compartment (L/h)
    ksr = 0.006
    krs = 3/2*ksr
    Vb = model.Vs[t] + model.Vrbc[t]
    kes = 4
    kse = 11/3*kes + (1/(1+ 1*exp(-6*(Vb - 7))))*2.1429*model.Ve[t]
    Vtot = model.Vi[t]+model.Ve[t]+model.Vs[t]+model.Vrbc[t]
    S = 0.1/3 / (1+exp(-6*(Vtot - 38.2)))
    return model.dVs[t] == krs*model.Vrbc[t] - ksr*model.Vs[t] + kes*model.Ve[t] - kse*
        model.Vs[t] \
                           + IV_in(t) + Oral_in(t) + Cont_in(t) + Blood_in(t) + Oper_in(t)
                               + Feed_in(t) \
                           + Enteral_in(t) - model.kc[t] - S
model.Vs_eq = Constraint(model.t, rule=_Vs_eq)

def _Vrbc_eq(model,t):
    ksr = 0.006
    krs = 3/2*ksr
    return model.dVrbc[t] == ksr*model.Vs[t] - krs*model.Vrbc[t]
model.Vrbc_eq = Constraint(model.t, rule=_Vrbc_eq)

def _Ve_eq(model,t):
    kie = 5 / (1 + exp(3*(model.Vi[t] - 26)))
    kei = kie*26/11
    Vb = model.Vs[t] + model.Vrbc[t]
    kes = 4
    kse = 11/3*kes + (1/(1+ 1*exp(-6*(Vb - 7))))*2.1429*model.Ve[t]
    return model.dVe[t] == kie * model.Vi[t] + kse*model.Vs[t] - kei*model.Ve[t] - 4*
        model.Ve[t] - Ve_out(t)
model.Ve_eq = Constraint(model.t, rule=_Ve_eq)

def _Vi_eq(model,t):
```

```
    kie  = 5 / (1 + exp(3*(model.Vi[t] - 26)))
    kei  = kie*26/11
    return model.dVi[t] == kei*model.Ve[t] - kie*model.Vi[t]
model.Vi_eq = Constraint(model.t, rule=_Vi_eq)
"""
Creatinine Concentration Compartments
"""
def _Cs_eq(model,t):
    R = 3.12*(0.7*model.Vrbc[t]/((0.7/0.93)*(model.Vrbc[t]/(model.Vs[t]-model.Vrbc[t]))
        +1))
    kGFR = 44/(3*8)*model.GFR_s[t] ################## Operates off of
        initial conditions of Vs and Cs
    return model.dCs[t] == 5*(model.Cxm[t]+model.Cxe[t]-2*model.Cs[t])/model.Vs[t] +
        R(model.Crbc[t]-model.Cs[t])/model.Vs[t] - kGFR*model.Cs[t] - model.Cs[t]/
        model.Vs[t]*model.dVs[t]
model.Cs_eq = Constraint(model.t, rule=_Cs_eq)

def _Crbc_eq(model,t):
    R = 3.12*(0.7*model.Vrbc[t]/((0.7/0.93)*(model.Vrbc[t]/(model.Vs[t]-model.Vrbc[t]))
        +1))
    return model.dCrbc[t] == R*(model.Cs[t]-model.Crbc[t])/model.Vrbc[t] - model.Crbc
        [t]/model.Vrbc[t]*model.dVrbc[t]
model.Crbc_eq = Constraint(model.t, rule=_Crbc_eq)

def _Cxe_eq(model,t):
    Vxe  = 0.6*(model.Vi[t] + model.Ve[t])
    dVxe = 0.6*(model.dVi[t] + model.dVe[t])
    Q = 5
    return model.dCxe[t] == Q/Vxe*(model.Cs[t]-model.Cxe[t]) - model.Cxe[t]/Vxe*
        dVxe
model.Cxe_eq = Constraint(model.t, rule=_Cxe_eq)
"""
Creatinine Generation
"""
def _Cxm_eq(model,t):
    Vxm = 0.4*(model.Vi[t] + model.Ve[t])
    dVxm = 0.4*(model.dVi[t] + model.dVe[t])
    G = 30 + (52.9557 - 30)*model.GFR_s[t]/(0.6397 + model.GFR_s[t]) #model.G_min +
        (model.G_max)*(model.GFR_s[t])/(model.k_G + model.GFR_s[t])
    Q = 5
    return model.dCxm[t] == G/Vxm + Q*(model.Cs[t] - model.Cxm[t])/Vxm - model.
        Cxm[t]/Vxm*dVxm
model.Cxm_eq = Constraint(model.t, rule=_Cxm_eq)
"""
Define Objective Function
"""
time = return_pairs(np.array(model.t))
```

```
time_2 = Urineout.index.values
time_3 = df_SCr.index.values

def objectivefunc(model):
    return sum(50*[((model.GFR_s[time[n][1]]-model.GFR_s[time[n][0]])/(time[n][1]-time[n
        ][0]))**2 for n in range(0,len(df1.index.values[1:]))]) \
        + sum(1*[((model.kc[time[p][1]] - model.kc[time[p][0]])/(time[p][0] - time[p][1]))
            **2 for p in range(0,len(df1.index.values[1:])) ]) \
        + sum(1100*[(model.B_tot[time_2[z]] - Urineout.loc[time_2[z], 'Urine Tot'])**2 for
            z in range(len(time_2))]) \
        + sum(700*[(df_SCr.loc[time_3[y], 'SCr'] - model.Cs[time_3[y]])**2 for y in range(
            len(df_SCr.index.values))]) \
        + sum(1*[((model.Cs[time[w-1][1]]-model.Cs[time[w-1][0]])/(model.t[w+1] -
            model.t[w]))**2 for w in range(1,len(df1.index.values[1:]))])

model.objective = Objective(rule=objectivefunc)
"""
SOLVING
"""
# Solver options
NFE = len(model.t)*1
MaxIter = 10000
TOL = 1e-6
## Boilerplate code to solve -- converts ODEs into algebraic equations
discretizer =TransformationFactory("dae.finite_difference") #discretizes the entire model
discretizer.apply_to(model,nfe=NFE,wrt=model.t,scheme="BACKWARD") # discretizes
    the entire model
# solve the problem
opt = SolverFactory('ipopt')
opt.options['linear_solver'] = "ma97"
opt.options['tol'] = TOL
opt.options['max_iter'] = MaxIter

"""
RESULTS
"""
results = opt.solve(model, keepfiles=False, tee=True)
model.solutions.load_from(results)
print(results.solver.termination_condition)
W = '\033[0m' # white (normal)
R = '\033[31m' # red
G = '\033[32m' # green
O = '\033[33m' # orange
B = '\033[34m' # blue
P = '\033[35m' # purple
print(B+'Patient %s' %(NUMBER)+W)
```

```
gfrs = np.array([model.GFR_s[t]() for t in df1.index.values])
Cs = np.array([model.Cs[t]() for t in df1.index.values])
Crbc = np.array([model.Crbc[t]() for t in df1.index.values])
Cxe = np.array([model.Cxe[t]() for t in df1.index.values])
Cxm = np.array([model.Cxm[t]() for t in df1.index.values])
Vi = np.array([model.Vi[t]() for t in df1.index.values])
Vs = np.array([model.Vs[t]() for t in df1.index.values])
Vrbc = np.array([model.Vrbc[t]() for t in df1.index.values])
Ve = np.array([model.Ve[t]() for t in df1.index.values])
kc = np.array([model.kc[t]() for t in df1.index.values])
B_tot = np.array([model.B_tot[t]() for t in df1.index.values])
FI_tot = np.array([model.FI_tot[t]() for t in df1.index.values])
FO_tot = np.array([model.FO_tot[t]() for t in df1.index.values])

"""
Model Optimized Parameters
"""
kc_vol = model.kc_vol()
kc_max = model.kc_max()
kc_slope = model.kc_slope()
kc_min = model.kc_min()

np.savez("Final_Results_edr_%d" %(NUMBER), gfrs, Cs, Crbc, Cxe, Cxm, Vi, Vs, Vrbc,Ve,
    kc, B_tot, kc_min, kc_max, kc_slope, kc_vol)
clear()
```

APPENDIX E

THEORETICAL PATIENTS

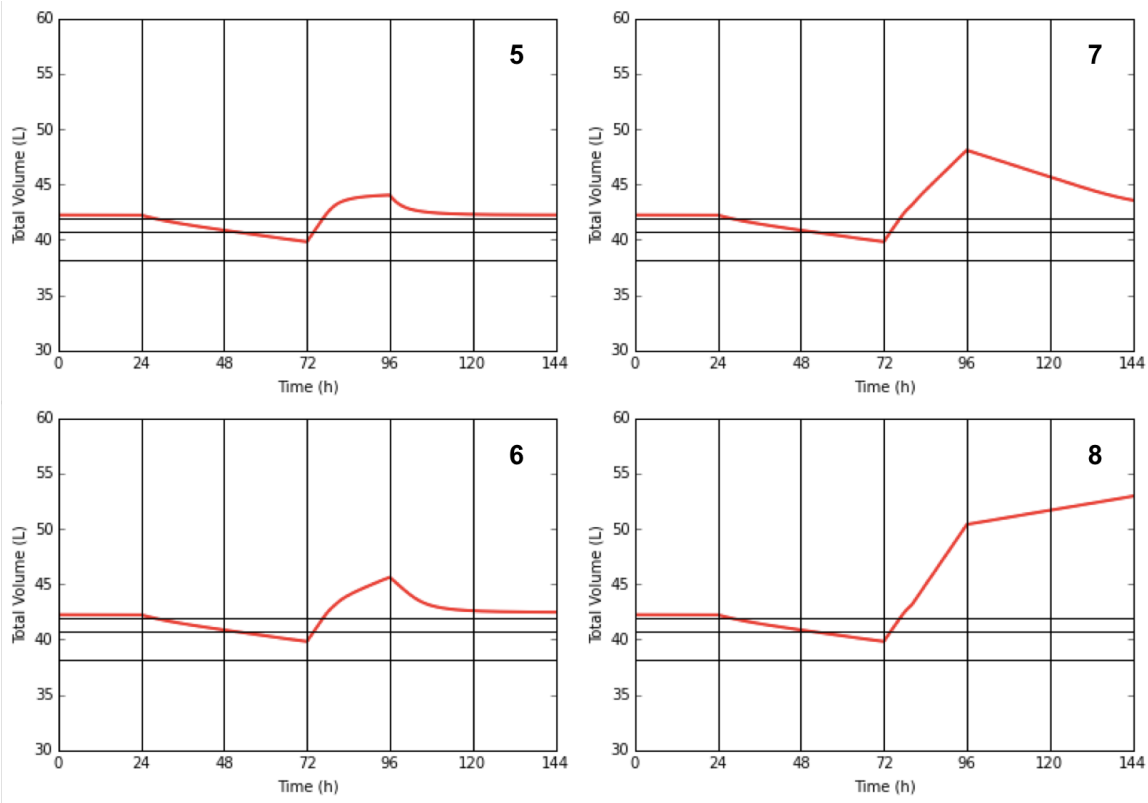

Figure E1: Theoretical patients (5-8) resulting volumes when baseline SCr is 12mgL, which will not allow a return to 100% kidney function and nominal k_c. Black, horizontal lines represent 100% hydration, minor dehyrdation, and severe dehydration in descending order. Initial volume distribution set by referencing built library for initial conditions to give equilibrium at t=0h. Patient number is given in the upper, right-hand corner of each subplot.

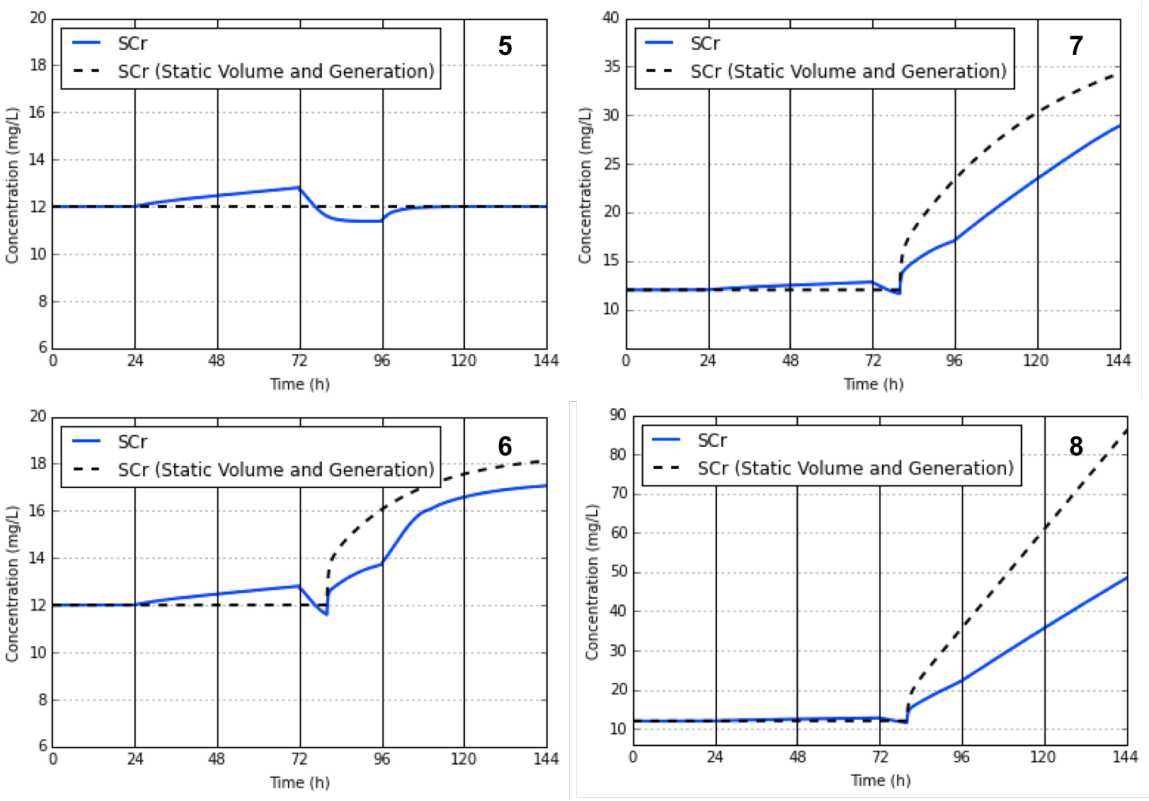

Figure E2: Theoretical patients SCr over the course of six days (144h). Baseline SCr is 12mgL^{-1} (Stage 2 CKD). Patient number is given in the upper, right-hand corner of each subplot.

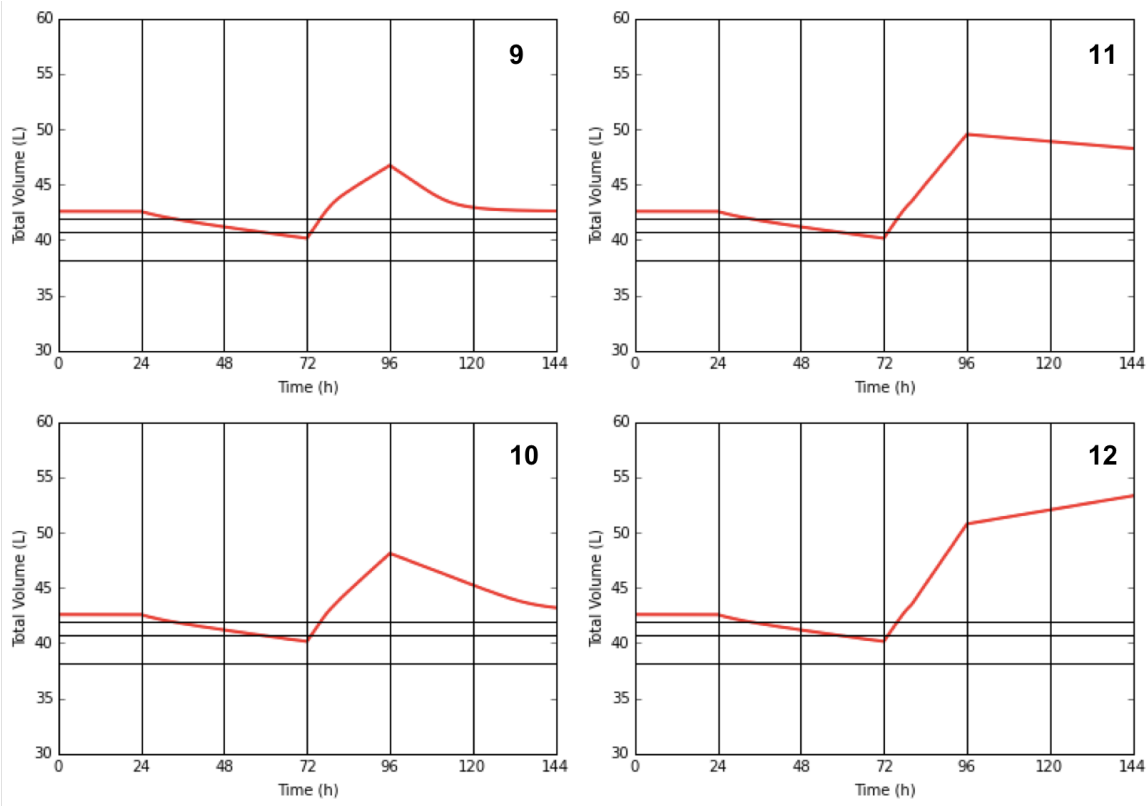

Figure E3: Theoretical patients (9-12) resulting volumes when baseline SCr is 21mgL, which will not allow a return to 100% kidney function and nominal k_c. Black, horizontal lines represent 100% hydration, minor dehyrdation, and severe dehydration in descending order. Initial volume distribution set by referencing built library for initial conditions to give equilibrium at t=0h. Patient number is given in the upper, right-hand corner of each subplot.

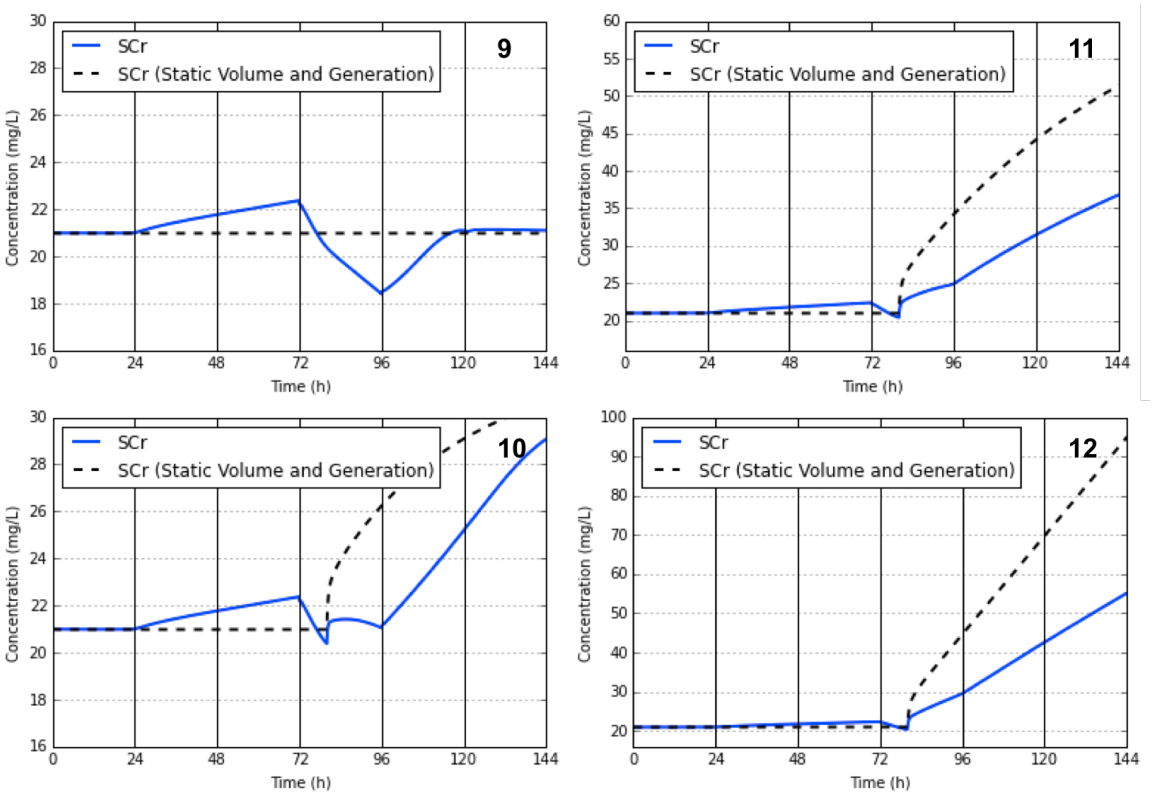

Figure E4: Theoretical patients SCr over the course of six days (144h). Baseline SCr is 21mgL^{-1} (Stage 3 CKD). Patient number is given in the upper, right-hand corner of each subplot.

BIBLIOGRAPHY

[1] R. Bellomo, C. Ronco, J. A. Kellum, R. L. Mehta, P. Palevsky, and t. A. Workgroup. Acute renal failure definition, outcome measures, animal models, fluid therapy and information technology needs: the Second International Consensus Conference of the Acute Dialysis Quality Initiative (ADQI) Group. *Critical Care*, 8:R204, 2004.

[2] T. Chatterjee, B. K. Chatterjee, D. Majumdar, and P. Chakrabarti. Antibacterial effect of silver nanoparticles and the modeling of bacterial growth kinetics using a modified Gompertz model. *Biochimica et Biophysica Acta - General Subjects*, 1850(2):299–306, 2015.

[3] W. R. Clark, B. A. Mueller, M. A. Kraus, and W. L. Macias. Quantification of creatinine kinetic parameters in patients with acute renal failure. *Kidney International*, 54(2):554–560, 1998.

[4] K. Doi, P. S. Yuen, C. Eisner, X. Hu, A. Leelahavanichkul, J. Schnermann, and R. A. Star. Reduced Production of Creatinine Limits Its Use as Marker of Kidney Injury in Sepsis. *Journal of the American Society of Nephrology*, 20(6):1217–1221, 2009.

[5] a. Dunn, V. Lo, and S. Donnelly. The role of the kidney in blood volume regulation: the kidney as a regulator of the hematocrit. *Am J Med Sci*, 334(1):65–71, 2007.

[6] L. M. Ebah. *Extraction and Analysis of Interstitial Fluid, and Characterisation of the Interstitial Compartment in Kidney Disease*. Phd, University of Manchester, 2012.

[7] P. C. Farrell, L. G. Norman, L. F. DeLoss, R. P. Popovich, J. W. Broviac, and A. L. Babb. A comparison of in vitro and in vivo solute-protein binding interactions in normal and uremic subjects. *Transactions - American Society for Artificial Internal Organs*, 18:268, 1972.

[8] A. C. Fry and K. Farrington. Management of acute renal failure. *Postgrad Med J. Feb*, 82(964):106–116, 2006.

[9] S. R. Gadagkar and G. B. Call. Computational tools for fitting the Hill equation to dose-response curves. *Journal of Pharmacological and Toxicological Methods*, 71:68–76, 2015.

[10] K. Goddard and S. J. Austin. Appropriate regulation of routine laboratory testing can reduce the costs associated with patient stay in intensive care, 2011.

[11] R. Gray and F. Baldwin. Targeting blood tests in the ICU may lead to a significant cost reduction. *Intensive Care Med*, 28, 2002.

[12] L. J. Guo, B. Zhao, Q. An, and M. Tian. Characteristics of a Novel Aerobic Denitrifying Bacterium, Enterobacter cloacae Strain HNR. *Applied Biochemistry and Biotechnology*, 178(5):947–959, 2016.

[13] A. C. Guyton and J. E. Hall. *Textbook of Medical Physiology*. W.B. Saunders Company, Philadelphia, 10th edition, 2000.

[14] K. Hallow and Y. Gebremichael. A Quantitative Systems Physiology Model of Renal Function and Blood Pressure Regulation: Model Description. *CPT: Pharmacometrics & Systems Pharmacology*, 6(6):383–392, 2017.

[15] W. E. Hart. Python optimization modeling objects (Pyomo). In *Operations Research and Cyber-Infrastructure*, pages 3–19. Springer, 2009.

[16] O. Heimbürger, P. Stenvinkel, and P. Bárány. The enigma of decreased creatinine generation in acute kidney injury. *Nephrology Dialysis Transplantation*, 27(11):3973–3974, 2012.

[17] E. A. Hoste, G. Clermont, A. Kersten, R. Venkataraman, D. C. Angus, D. De Bacquer, and J. A. Kellum. RIFLE criteria for acute kidney injury are associated with hospital mortality in critically ill patients: a cohort analysis. *Critical Care*, 10(3):R73, 2006.

[18] J. D. Hunter. Matplotlib: a 2D Graphics Environment, 2007.

[19] IOM. Water. In *Dietary Reference Intakes for water, potassium, sodium, chloride and sulfate*, chapter 4. National Academies Press, 2004.

[20] Joint United Kingdom Blood Transfusion and Tissue Transplantation Services Professional Advisory. *Handbook of Transfusion Medicine*. The Stationary Office, 5th edition, 2013.

[21] J. Kim, Z. Wang, S. B. Heymsfield, R. N. Baumgartner, and D. Gallagher. Total-body skeletal muscle mass : estimation by a new dual-energy. *The American journal of clinical nutrition*, 76:378–383, 2002.

[22] E. K. Lee, T. Fox, and I. Crocker. Simultaneous Beam Geometry and Intensity Map Optimization in Intensity-Modulated Radiation Therapy. *Int. J. Radiation Oncology Biol. Phys.*, 64:301–320, 2006.

[23] E. Macedo, R. Malhotra, J. e. e. Bouchard, S. K. Wynn, and R. L. Mehta. Oliguria is an early predictor of higher mortality in critically ill patients. *Kidney International*, 80(7):760–767, 2011.

[24] H. Matsui, B. R. Grubb, R. Tarran, S. H. Randell, J. T. Gatzy, C. W. Davis, and R. C. Boucher. Evidence for periciliary liquid layer depletion, not abnormal ion composition, in the pathogenesis of cystic fibrosis airways disease. *Cell*, 95:1005–1015, 1998.

[25] P. a. Mccullough, J. Bouchard, S. S. Waikar, E. D. Siew, H. Zoltan, S. L. Goldstein, J. L. Koyner, E. Macedo, K. Doi, and S. Di. NIH Public Access. *Contributions to nephrology*, 48374:1–8, 2013.

[26] W. McKinney. Data Structures for Statistical Computing in Python, 2011.

[27] M. T. Nguyen and P. Devarajan. Biomarkers for the early detection of acute kidney injury. *Pediatric Nephrology*, 23(12):2151–2157, 2008.

[28] NIDDK. Kidney Disease Statistics for the United States, 2016.

[29] S. F. Okada, R. A. Nicholas, S. M. Kreda, E. R. Lazarowski, and R. C. Boucher. Physiological regulation of ATP release at the apical surface of human airway epithelia. *Journal of Biological Chemistry*, 281(32):22992–23002, 2006.

[30] M. Ostermann and M. Joannidis. Biomarkers for AKI improve clinical practice: no. *Intensive Care Medicine*, 41(4):618–622, 2015.

[31] G. Pesola, I. Akhavan, and C. Carlon. Urinary creatinine excretion in the ICU: low excretion does not mean inadequate collection. *Am J Crit. Care*, 2(6):462–466, 1993.

[32] J. W. Pickering, E. Author, A. M. Ralib, and Z. H. Endre. Combining creatinine and volume kinetics identifies missed cases of acute kidney injury following cardiac arrest. *Critical Care*, 17(1), 2013.

[33] J. R. Prowle. Acute kidney injury: Creatinine and AKI-through a glass, darkly. *Nat Rev Nephrol*, 9(4):193–195, apr 2013.

[34] J. R. Prowle. Measurement of AKI biomarkers in the ICU: still striving for appropriate clinical indications. *Intensive Care Medicine*, 41(3):541–543, 2015.

[35] R. Rockenfeller and M. Günther. Hill equation and Hatze's muscle activation dynamics complement each other: enhanced pharmacological and physiological interpretability of modelled activity-pCa curves. *Journal of Theoretical Biology*, 431:11–24, 2017.

[36] C. Ronco, A. Levin, D. G. Warnock, R. Mehta, J. A. Kellum, S. Shah, B. A. Molitoris, and A. W. Group. Improving outcomes from acute kidney injury (AKI): Report on an initiative. *Int J Artif Organs*, 30(5):373–376, 2007.

[37] M. Schetz and E. Hoste. Understanding oliguria in the critically ill. *Intensive Care Medicine*, 43(6):914–916, 2017.

[38] D. Schneditz, D. Platzer, and J. T. Daugirdas. A diffusion-adjusted regional blood flow model to predict solute kinetics during haemodialysis. *Nephrology Dialysis Transplantation*, 24(7):2218–2224, 2009.

[39] D. Schneditz, Y. Yang, G. Christopoulos, and J. Kellner. Rate of creatinine equilibration in whole blood. *Hemodialysis International*, 13:215–221, 2009.

[40] C. K. Sinha and M. Davenport. *Handbook of pediatric surgery*. London, 2010.

[41] L. A. Stevens, M. A. Claybon, C. H. Schmid, M. Horio, E. Imai, R. G. Nelson, M. V. Deventer, H.-y. Wang, L. Zuo, Y. L. Zhang, and A. S. Levey. Evaluation of the CKD-EPI Equation in Multiple Races and Ethnicities. *Kidney International*, 79(5):555–562, 2011.

[42] S. van der Walt, C. Colbert, and G. Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation, 2011.

[43] A. Wachter and L. T. Biegler. On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming*, 106(1):25–57, 2006.

[44] S. S. Waikar and J. V. Bonventre. Creatinine Kinetics and the Definition of Acute Kidney Injury. *J Am Soc Nephrol Mar*, 20(3):672–679, 2009.

[45] F. P. Wilson, J. M. Sheehan, L. H. Mariani, and J. S. Berns. Creatinine generation is reduced in patients requiring continuous venovenous hemodialysis and independently predicts mortality. *Nephrology Dialysis Transplantation*, 27(11):4088–4094, 2012.